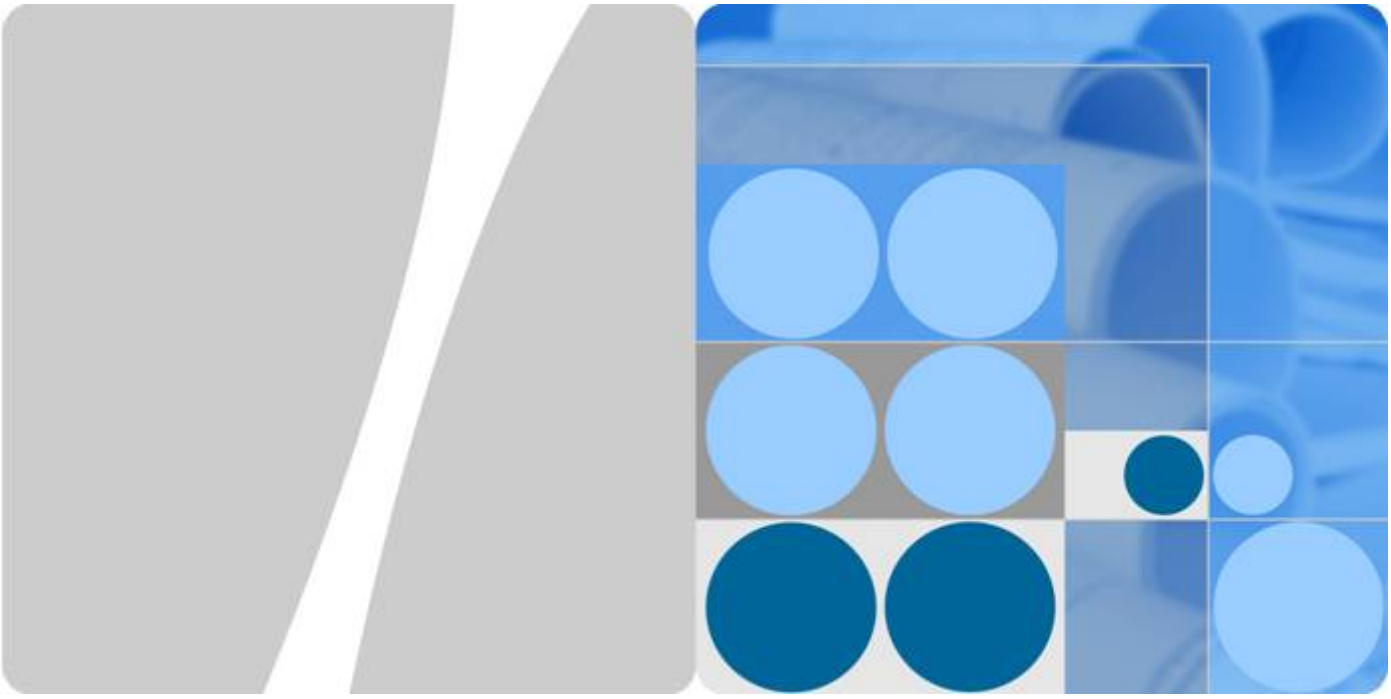


Huawei EulerOS V2.0 Administrators Guide

Issue 03

Date 2019-08-14

HUAWEI TECHNOLOGIES CO., LTD.

HUAWEI

Trademarks and Permissions


and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: http://www.huawei.com

Email: support@huawei.com

Contents

1 Preface

Overview

Huawei EulerOS V2.0 is the Huawei Enterprise Linux server operating system. It is easy to maintain, compatible with mainstream software and hardware, and exhibits high performance, reliability, and security.

This guide describes how to use, configure, and maintain Huawei EulerOS V2.0.

Intended Audience

This guide is intended for Huawei EulerOS V2.0 users with a basic understanding of Linux system management, and is also recommended for:

- Administrators
- System engineers
- Maintenance personnel

2 Basic Configuration

2.1 Using Commands

2.1.1 Setting the System Locale

System locale settings are stored in the /etc/locale.conf file and can be modified by the localectl command. These settings are read at early boot by the systemd daemon.

Displaying the Current Locale Status

To display the current locale status, run the following command:

```
localectl status
```

Example command output:

```
$ localectl status
  System Locale: LANG=zh_CN.UTF-8
        VC Keymap: cn
       X11 Layout: cn
```

Listing Available Locales

To list available locales, run the following command:

```
localectl list-locales
```

Imagine you want to select a specific Chinese locale, but you are not sure if it is available on EulerOS. You can check that by listing all Chinese locales with the following command:


```
$ localectl list-locales | grep zh
 zh_CN
 zh_CN.gb18030
 zh_CN.gb2312
 zh_CN.gbk
 zh_CN.utf8
 zh_HK
 zh_HK.big5hkscs
 zh_HK.utf8
 zh_SG
 zh_SG.gb2312
 zh_SG.gbk
 zh_SG.utf8
 zh_TW
 zh_TW.big5
 zh_TW.euctw
 zh_TW.utf8
```

Setting the Locale

To set the locale, run the following command as the root user:

```
localectl set-locale LANG=locale
```

Replace *locale* with the locale name you want to use. For example, if you want to use Simplified Chinese as the locale, run the following command as the root user:

```
# localectl set-locale LANG=zh_CN.utf8
```

2.1.2 Setting the Keyboard Layout

Keyboard layout settings are stored in the /etc/locale.conf file and can be modified by the localectl command. These settings are read at early boot by the systemd daemon.

Displaying the Current Settings

To display the current keyboard layout settings, run the following command:

```
localectl status
```

Example command output:

```
$ localectl status
  System Locale: LANG=zh_CN.UTF-8
        VC Keymap: cn
       X11 Layout: cn
```

Listing Available Keyboard Layouts

To list all available keyboard layouts that can be configured on EulerOS, run the following command:

```
localectl list-keymaps
```

To list keyboard layouts compatible with your current locale (for example, Chinese), run the following command:

```
$ localectl list-keymaps | grep cn
 cn
```

Setting the Keyboard Layout

To set the keyboard layout, run the following command as the root user:


```
localectl set-keymap map
```

The keyboard layout will be equally applied to graphical user interfaces.

Then you can verify if your setting was successful by checking the current status:

```
$ localectl status
  System Locale: LANG=zh_CN.UTF-8
        VC Keymap: cn
       X11 Layout: us
```

2.1.3 Setting the Date and Time

This topic describes how to set the system date, time, and time zone by using timedatectl, date, and hwclock commands.

2.1.3.1 Using the timedatectl Command

Displaying the Current Date and Time

To display the current date and time, run the following command:

```
timedatectl
```

Example command output:

```
$ timedatectl
       Local time: 2015-08-14 15:57:24 CST
   Universal time: 2015-08-14 07:57:24 UTC
         RTC time: 2015-08-14 07:57:24
         Timezone: Asia/Shanghai (CST, +0800)
      NTP enabled: yes
 NTP synchronized: no
  RTC in local TZ: no
       DST active: n/a
```

Changing the Current Time

To change the current time, run the following command as the root user:

```
timedatectl set-time HH:MM:SS
```

For example, to change the current time to 15:57:24 pm, run the following command as the root user:

```
# timedatectl set-time 15:57:24
```

Changing the Current Date

To change the current date, run the following command as the root user:

```
timedatectl set-time YYYY-MM-DD
```

For example, to change the current date to 14 August 2015, run the following command as the root user:

```
# timedatectl set-time '2015-08-14'
```

Changing the Time Zone

To list all available time zones, run the following command:

```
timedatectl list-timezones
```

To change the current time zone, run the following command as the root user:

```
timedatectl set-timezone time_zone
```

Imagine you want to identify which time zone is closest to your present location while you are in Asia. You can check that by listing all available time zones in Asia with the following command:

```
# timedatectl list-timezones | grep Asia
 Asia/Aden
 Asia/Almaty
 Asia/Amman
 Asia/Anadyr
 Asia/Aqtau
 Asia/Aqtobe
 Asia/Ashgabat
 Asia/Baghdad
 Asia/Bahrain
 ……
 Asia/Seoul
 Asia/Shanghai
 Asia/Singapore
 Asia/Srednekolymsk
 Asia/Taipei
 Asia/Tashkent
 Asia/Tbilisi
 Asia/Tehran
 Asia/Thimphu
 Asia/Tokyo
```

To change the time zone to shanghai, run the following command:

```
# timedatectl set-timezone Asia/Shanghai
```

Synchronizing the System Clock with a Remote Server

Your system clock can be automatically synchronized with a remote server using the Network Time Protocol (NTP). To enable or disable this feature, run the following command as the root user:

```
timedatectl set-ntp boolean
```

For example, to automatic synchronization of the system clock with a remote server, run the following command:

```
# timedatectl set-ntp yes
```

2.1.3.2 Using the date Command

Displaying the Current Date and Time

To display the current date and time, run the following command:

```
date
```

By default, the date command displays the local time. To display the time in Coordinated Universal Time (UTC), run the command with the --utc or -u command line option:

```
date --utc
```

You can also customize the format of the displayed information by providing the + "format" option on the command line:


```
date +"format"
```

Table 2-1 Formatting options

Format Option	Description
%H	The hour in the HH format (for example, 17)
%M	The minute in the MM format (for example, 37)
%S	The second in the SS format (for example, 25)
%d	The day of the month in the DD format (for example, 15)
%m	The month in the MM format (for example, 07)
%Y	The year in the YYYY format (for example, 2015)
%Z	The time zone abbreviation (for example, CEST)
%F	The full date in the YYYY-MM-DD format (for example, 2015-7-15). This option is equal to %Y-%m-%d.
%T	The full time in the HH:MM:SS format (for example, 18:30:25). This option is equal to %H:%M:%S.

Example commands and outputs:

- To display the current date and time:
  ```
  $ date
   2015 08 17 Monday 17:26:34 CST
  ```
- To display the current date and time in UTC:
  ```
  $ date --utc
   2015 08 17 Monday 09:26:18 UTC
  ```
- To customize the output of the date command:
  ```
  $ date +"%Y-%m-%d %H:%M"
   2015-08-17 17:24
  ```

Changing the Current Time

To change the current time, run the date command with the --set or -s option as the root user:

```
date --set HH:MM:SS
```

By default, the date command sets the local time. To set the system clock in UTC instead, run the command with the --utc or -u command line option:

```
date --set HH:MM:SS --utc
```

For example, to change the current time to 23:26:00 p.m, run the following command as the root user:

```
# date --set 23:26:00
```

Changing the Current Date

To change the current date, run the date command with the --set or -s option as the root user:

```
date --set YYYY-MM-DD
```

For example, to change the current date to 2 November 2015, run the following command as the root user:

```
# date --set 2015-11-02
```

2.1.3.3 Using the hwclock Command

The hwclock command is used to set the real-time clock (RTC).

Real-Time Clock and System Clock

Linux distinguishes between the system clock and the real-time clock. The system clock is maintained by Linux kernel, whereas the real-time clock is an integrated clock on the system board and is battery-powered. The real-time clock is defined in the Standard BIOS Feature option of BIOS.

Once Linux is booted, the real-time clock reads system clock settings and then the system clock is completely independent of the real-time clock.

All commands in the system, including functions, are based on the system clock. The system clock does not conflict with the real-time clock, and the two can even be different. This is of little help to common users, but is greatly enjoyed by Linux network administrators. Imagine a network covers multiple time zones and files on server A in time zone A need to be synchronized to server B in time zone B. The network administrator can merely change the system clock on server B to time zone A without changing the real-time clock on that server. After the file synchronization is completed, the administrator can change the system clock on server B back to time zone B.

Displaying the Current Date and Time

To display the current RTC date and time, run the following command as the root user:

```
hwclock
```

Example command output:

```
# hwclock
 2015-08-17  Monday 14:34:42. -0.094973 second
```

Setting the Date and Time

To change the RTC date and time, run the following command as the root user:

```
hwclock --set --date "dd mmm yyyy HH:MM"
```

For example, to change the RTC time to 21:17, 21 October 2015, run the following command as the root user:


```
# hwclock --set --date "21 Oct 2015 21:17" --utc
```

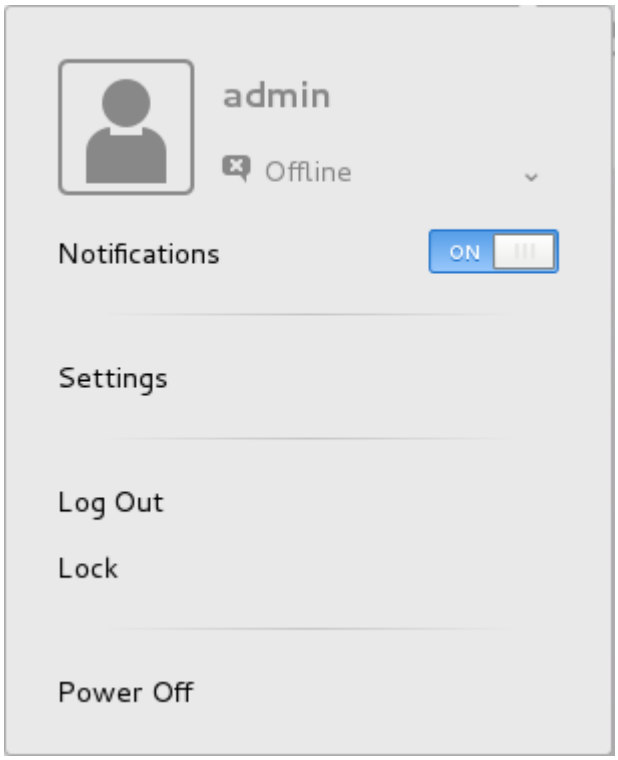
2.2 Using GUI

This topic guides you through configuring basic system options on graphical user interface (GUI).

2.2.1 Opening the Settings Page

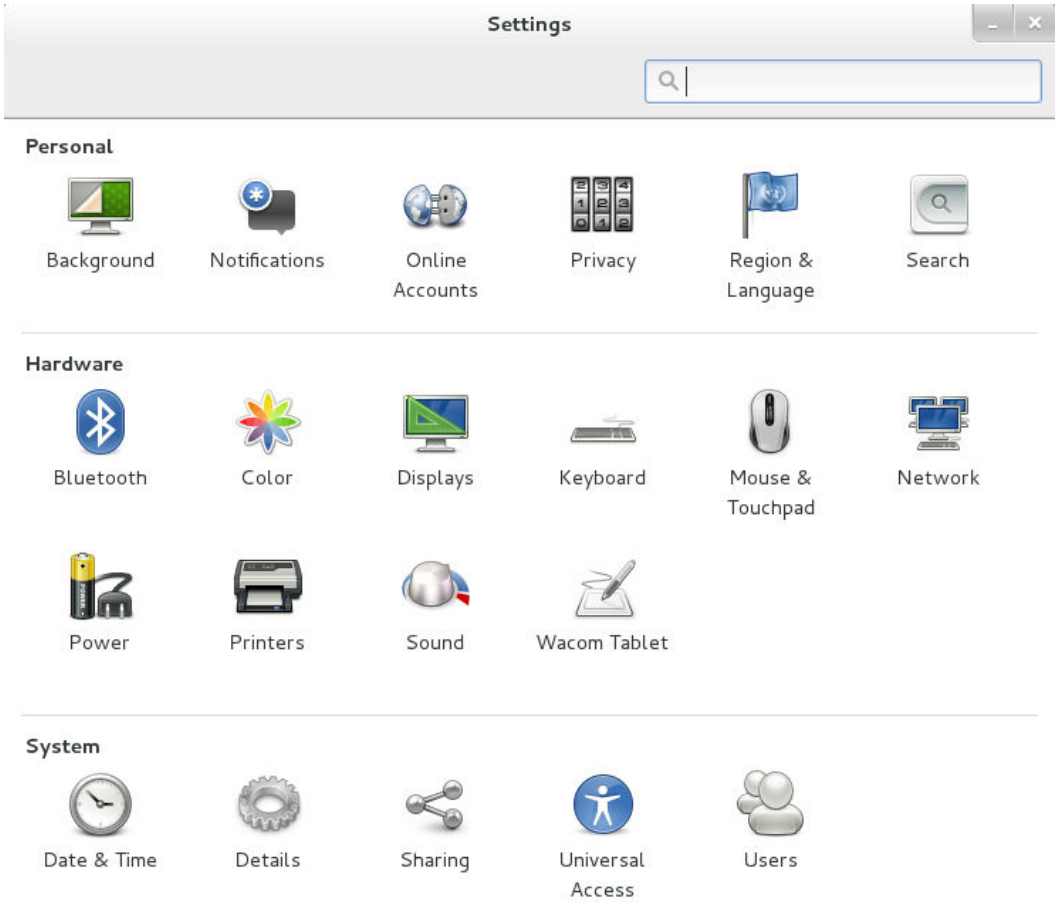
Click the user name in the upper right corner of desktop. On the displayed page, click **Settings**, as shown in **Figure 2-1**.

Figure 2-1 Opening the Settings page



The Settings page is displayed, as shown in **Figure 2-2**.

Figure 2-2 Settings page



2.2.2 Setting the Language

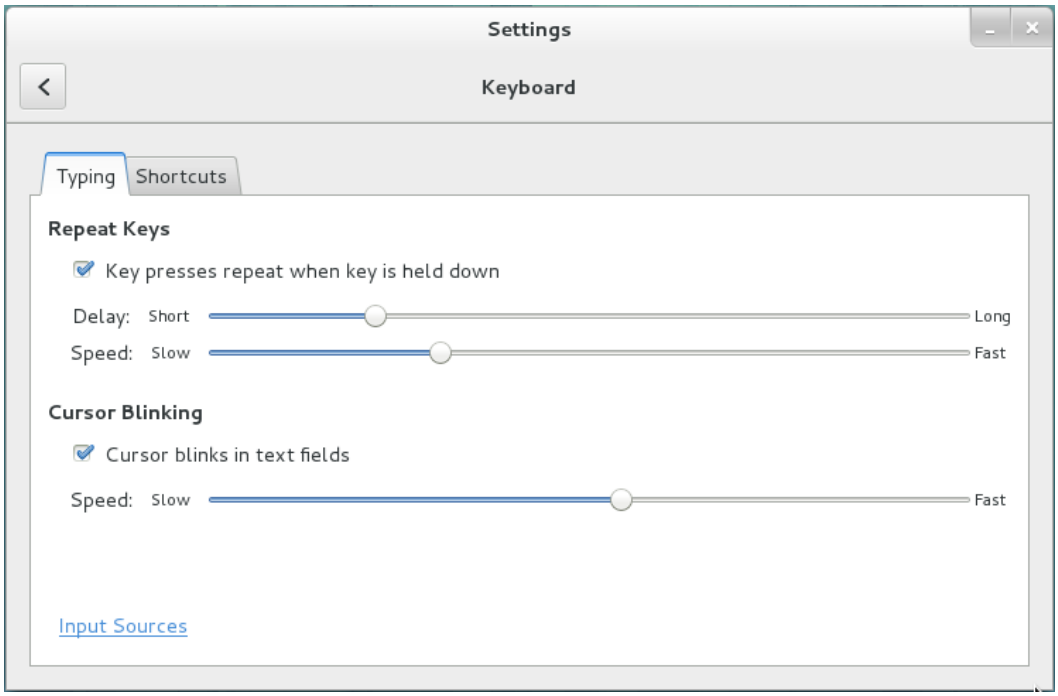
Click **Region & Language** in the **Personal** area of the **Settings page**. The **Region & Language** page is displayed, as shown in **Figure 2-3**.

Figure 2-3 Region & Language page

2.2.3 Setting the Keyboard Layout

Click **Keyboard** in the **Hardware** area of the **Settings page**. The **Keyboard** page is displayed, as shown in **Figure 2-4**.

Figure 2-4 Keyboard page

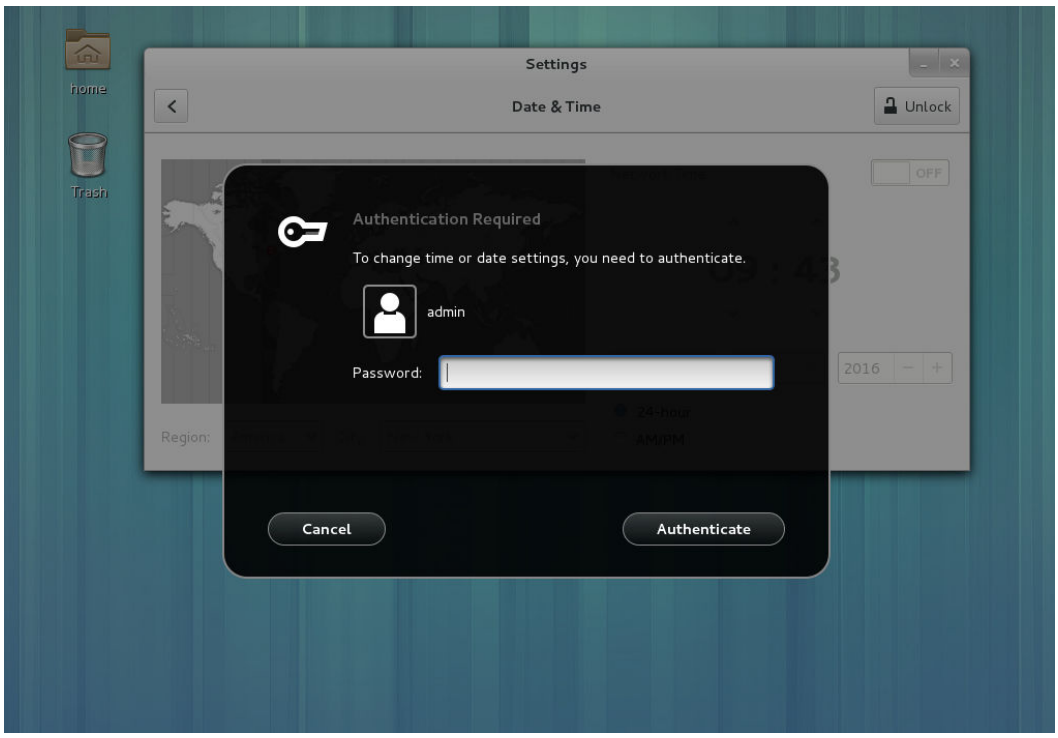


2.2.4 Setting the Date and Time

Click **Date & Time** in the **System** area of the **Settings page**. The **Date & Time** page is displayed.

If you have logged in to EulerOS as a non-root user, before setting the date and time, you must click **Unlock** in the upper right corner of the **Date & Time** page and then type the admin password (or the root user password if no admin user is created at the time of OS installation), as shown in **Figure 2-5**.

Figure 2-5 Authentication Required page



After authentication, you can set the region, city, and time, as shown in **Figure 2-6**.

Figure 2-6 Date & Time page

3 User Management

In Linux, each common user has an account, including the user name, password, and home directory. There also exist special users created for specific purposes, and the most important special user is the admin account whose default user name is root. The concept of user group is introduced to make privilege management easier. Each user belongs to at least one user group.

The control of users and groups is a core element of EulerOS security management. This topic explains how to create multiple admin accounts and assign privileges to common users in graphical user interface and on command lines.

3.1 Adding a User

useradd Command

The useradd command is used to add a new user to EulerOS.

Table 3-1 describes common useradd command options.

Table 3-1 Common useradd command options

Option	Description
-c comment	User's password file comment.

Option	Description
-d home dir	User's login directory. The default directory is /home/*account name*.
-e expire_date	Date (MM/DD/YY) on which the user's account will expire.
-f inactive_days	Number of days after the user's account expires until it is permanently disabled. If this option takes the value 0, the account is disabled immediately after it expires. If this option takes the value -1, the account is not be disabled after it expires.
-g initial_group	Group name or group number of the user's initial group.
-G	List of supplementary groups of the new account.
-M	Create the home directory automatically if it does not exist.
-n	Cancel the default settings in which the user group name is the same as the user name.
-r	Create a system account with a UID less than 500 and without a home directory. **NOTE** Use the -m option if you want to create a system account with a home directory.
-s shell	User's login shell.
-u uid	User ID of the user, which must be greater than 99. This value must be unique, unless the -o option is used. The value must be non-negative. Values 0 – 99 are reserved for system accounts and should not be assigned to users.
-D	Default useradd command option settings are displayed.

User Information Files

The following files contain user account information:

- /etc/passwd: user account information
- /etc/shadow file: user account encryption information
- /etc/group file: group information

- /etc/defaut/useradd: default configurations
- /etc/login.defs: system wide settings
- /etc/skel: default directory that holds initial configuration files

Example

To create a user XXX, run the following command:

```
[root@localhost ~]# useradd XXX
```

NOTE

If no prompt is displayed, the user XXX is successfully created. After the user XXX is created, run the passwd command to assign a password to the user. A new account without a password will be banned.

To view information about the new user, run the id command:

```
[root@localhost ~]#  id  user_example
 uid=502(user_example)    gid=502(user_example)
```

To change the password of the user_example, run the following command:

```
[root@localhost ~]# passwd  user_example
```

Then, type the password and confirm it as prompted:

```
Changing password for user user_example.
 New  password:
 BAD PASSWORD:it is based on a dictionary word
 Retype  new  password:
 passwd:all authentication tokens updated successfully.
```

3.2 Modifying a User Account

usermod Command

The usermod command is used to modify information in the system account file. Most options of the usermod command are the same as those of the useradd command. **Table 3-2** describes the options specific to the usermod command.

Table 3-2 Options specific to usermod

Option	Description
-a	Append the user to the supplemental GROUPS mentioned by the -G option without removing him/her from other groups.
-L	Locks the user's account.

Changing a Password

Common users can change their passwords using the passwd command. Only the admin is allowed to use the passwd usename command to change passwords for other users.

Changing User's Login Shell

Common users can change their login shell using either the chsh or usermod command. Only the admin is allowed to run the chsh usename command to change login shell for other users.

usermod command for changing user's shell:

```
usermod  -s  [new_shell_path] usename
```

For example, to change the shell of user_example to csh, run the following command:

```
[root@localhost ~]# usermod -s /bin/csh user_example
```

Changing the Home Directory

```
usermod  -d  [new_home_directory] usename
```

For example, to change the home directory of user_example to /home/user_example, run the following command:

```
[root@localhost ~]#usermod -d /home/user_example user_example
```

To move the content in the current home directory to a new one, run the usermod command with the -m option:

```
usermod  -d /new/home  -m  usename
```

Changing a UID

```
usermod  -u  UID  usename
```

The usermod command can change a user's UID in all files and directories under the user's home directory. However, for files outside the user's home directory, their owners can only be changed using the chown command.

Changing Account Expiry Date

To change account expiry date, run the following command. A prerequisite for using this command is that a shadow password is in use.

```
usermod  -e  MM/DD/YY  usename
```

3.3 Deleting a User

The userdel command is used to delete a user.

For example, to delete the user Test, run the following command:

```
[root@localhost ~]# userdel Test
```

If you need to also delete the user's home directory and all contents in the directory, run the userdel command with the -r option to delete them recursively.

NOTE

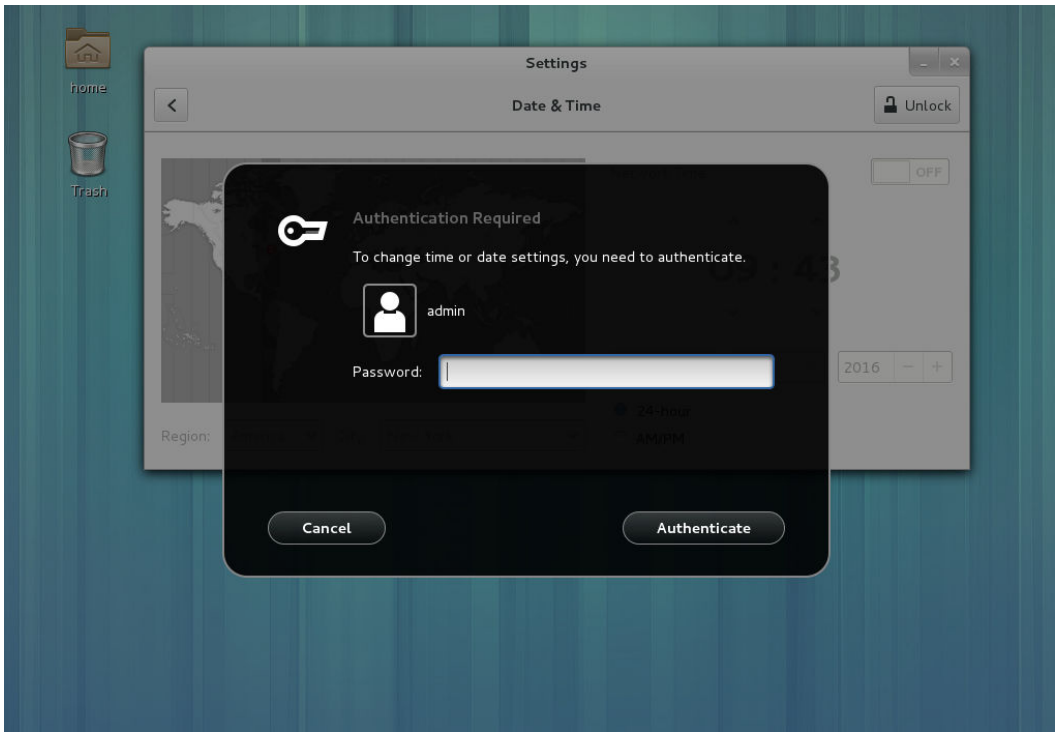
If a user has logged in to EulerOS, the user cannot be deleted unless you have first killed relevant processes.

3.4 Using GUI

Click **Users** in the **System** area of the **Settings page** to launch the **Users** page.

If you have logged in to EulerOS as a non-root user, before adding or deleting a user, you must click **Unlock** in the upper right corner of the **Users** page and then type the admin password (or the root user password if no admin user is created at the time of OS installation), as shown in **Figure 3-1**.

Figure 3-1 Authentication Required page



After the authentication is passed, click + or - in the lower left corner of the **Users** page to add or delete users.

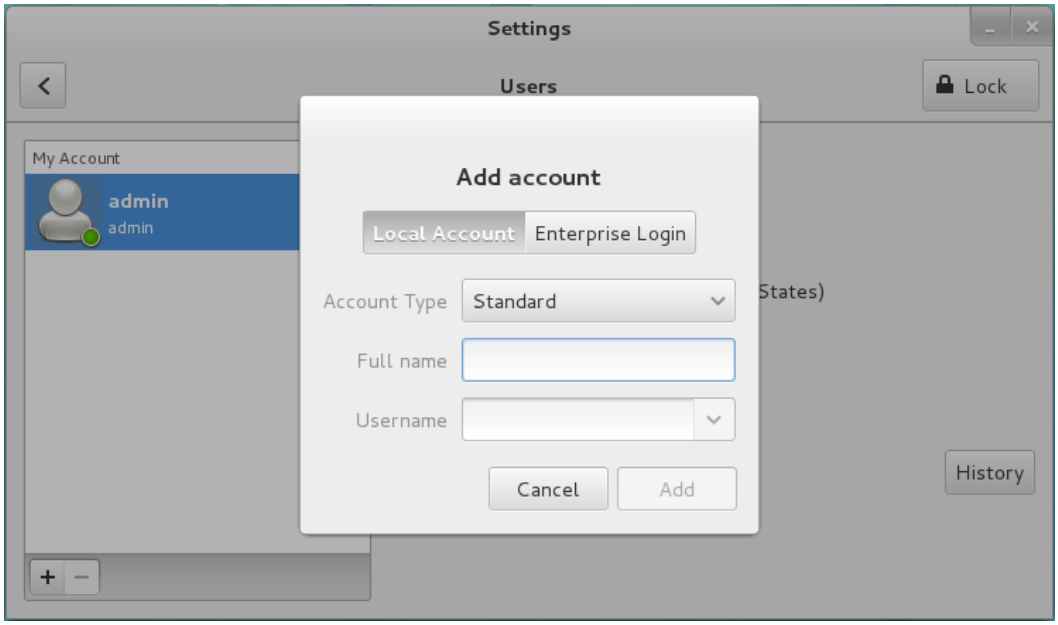
3.4.1 Adding a User

3.4.2 Deleting a User

3.4.1 Adding a User

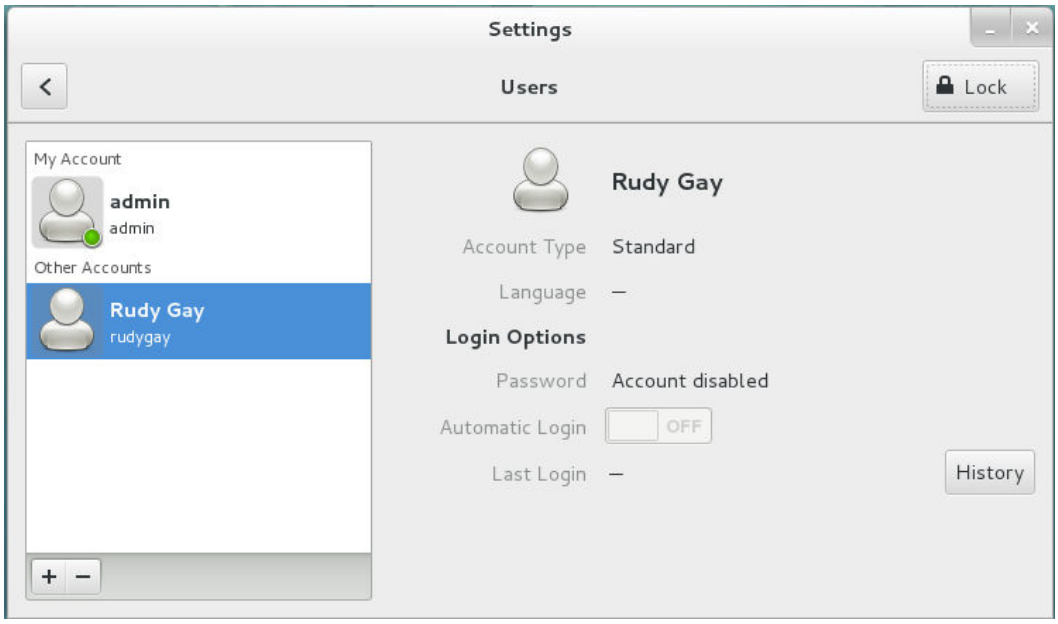
Click + in the lower left corner of the **Users** page to launch the **Add account** page, as shown in **Figure 3-2**.

Figure 3-2 Add account page



Specify the account type (standard or admin), full name, and user name. Then, click **Add**. Information about the added user is displayed, as shown in**Figure 3-3**.

Figure 3-3 Example user information



By default, the newly added user account is disabled and its login options need to be manually specified. To specify login options, click **Account disabled**, and on the displayed **Changing password** page, select the desired login option.

Figure 3-4 Changing the password for a newly added user

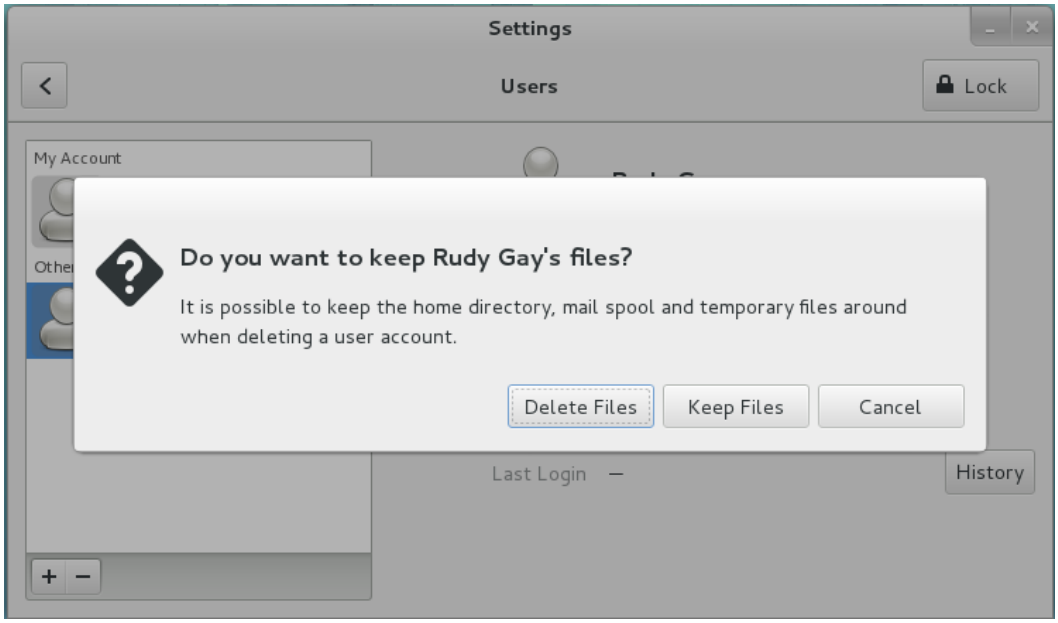
3.4.2 Deleting a User

Select the user you want to delete and click - in the lower left corner of the **Users** page. A message (see **Figure 3-5**) is then displayed, prompting you to confirm whether user files shall be deleted along with the user account.

NOTE

The user who is logging in to EulerOS cannot be deleted. Also, a common user is not allowed to delete the admin user if there is only one admin user.

Figure 3-5 Deleting a user



4 Software Package Management by Yum

Yum is a software package manager. Based on the RedHat package manager (RPM), Yum is able to automatically download RPM packages from specified servers (repositories) and install them. Yum performs automatic dependency management on packages you are updating, installing, or removing, and thus is able to automatically determine, fetch, and install all available dependent packages.

4.1 Configuring Yum

4.1.1 Modifying the Yum Configuration File

The main configuration file for yum is located at /etc/yum.conf. This file contains one mandatory [main] section, which allows you to set yum options that have global effect, and can also contain one or more [repository] sections, which allow you to set repository-specific options. Individual repositories are defined in .repo files in the /etc/yum.repos.d directory.

Configure Yum either using the yum.conf file under the /etc directory or by adding the .repo file under the /etc/yum.repos.d directory.

Setting the [Main] Section

The following is an example [main] section:

```
[main]
 cachedir=/var/cache/yum
 keepcache=0
 debuglevel=2
 logfile=/var/log/yum.log
 pkgpolicy=newest
 tolerant=1
 exactarch=1
 obsoletes=1
 gpgcheck=1
 plugins=1
 installonly_limit=3
 [comments abridged]
 # PUT YOUR REPOS HERE OR IN separate files named file.repo
 # in /etc/yum.repos.d
```

NOTE

For details about the complete [main] section, see yum.conf(5) in the online help of [main].

Table 4-1 Common options in the [main] section

Option	Description
cachedir=/var/cache/yum	Yum's cache directory where downloaded RPM packages and databases are saved. Imagine you want to install a software package. Yum will download the target package from the base/packages folder in the cache directory and install it automatically. Once installed, the target package will not be automatically deleted but it can be manually deleted.
keepcache=1	Determines whether Yum retains the cache of RPM packages and header files after a successful installation. Value: 0 or 1. The default value is 1 (retain the cache). If this option takes the value 1, the next time you want to install one of the RPM packages in the cache directory again, Yum will install it from the cache directory /var/cache/yum… without a need to first download it.
reposdir	Absolute path to the directory where .repo files are located. The default path is /etc/yum.repos.d.
assumeyes=0	Determines whether Yum automatically confirms RPM package installation. Value: 0 or 1. The default value is 0 (prompt for user confirmation).

Option	Description
alwaysprompt=1	Determines whether or not Yum always prompts for confirmation before installing an RPM package. Value: 0 or 1. The default value is 1 (always prompt for user confirmation).
retries=2	The number of retries following a network connection error. Setting this option to 0 makes Yum retry forever.
debuglevel=2	The level of details in the debug output produced by Yum. Value range: 1 – 10. A higher debug level causes Yum to display more detailed debug output.
logfile=/var/log/yum.log	Location of Yum log files.
pkgpolicy=newest	Software package selection policy in the event that multiple repositories hold the software package you want to install. Value: Newest (typical): Yum installs the latest version of the software package. Last: Yum places repository IDs in alphabetical order and installs the software package from the repository at the bottom of the alphabetical list.
distroverpkg=redhat-release	The package used by Yum to determine the version of the distribution. This can be any installed RPM package. The default package is redhat-release (for example, centos-release or rpmforge-release).
tolerant=1	Whether Yum is tolerant of package errors on command lines. Value: 0 or 1. The default value is 0 (not tolerant). Imagine you request to install 1.i386.rpm, 2.i386.rpm, and 3.i386.rpm packages but 3.i386.rpm is already installed. Setting this option to 1 prevents Yum from reporting the error that 3.i386.rpm is already installed.
exactarch=1	Determines whether Yum updates only the architectures of packages that you have installed. Value: 0 or 1. The default value is 1 (update architectures). For example, with the default value, Yum will not install an i686 package to update an i386 package.
obsoletes=1	This option only has affect during an update.

Option	Description
gpgcheck=1	Determines whether Yum performs a GPG signature check on packages. Value: 0 or 1. The default value is 1 (enable a GPG signature check).
plugins=1	A switch to enable or disable Yum plug-ins (for example, python). Value: 0 or 1 (enable).
metadata_expire=1800 exclude=….	The metadata_expire option specifies the time (in seconds) after which metadata will expire. The exclude option allows you to exclude packages by keyword during installation or updating. The value is either a wildcard (for example, * or ?) or a space-delimited list of software packages such as theme packages or particular patches.

Setting the [repository] Sections

The [repository] sections allow you to define individual Yum repositories. Each Yum repository must have a unique name. Otherwise, a conflict between repositories occurs. The following is a bare-minimum example of a [repository] section:

```
[repository]
 name=repository_name
 baseurl=repository_url
```

Table 4-2 Options in a [repository] section

Option	Description
name=repository_name	A string describing the repository.
baseurl=repository_url	Uniform resource locator (URL) to the directory where the repository is located. ● If the repository is available over Hypertext Transfer Protocol (HTTP), the URL is http://path/to/repo. ● If the repository is available over File Transfer Protocol (FTP), the URL is ftp://path/to/repo. ● If the repository is local to the machine, the URL is file:///path/to/local/repo.

Viewing Current Configurations

To display the current values of global Yum options, run the following command:

```
yum-config-manager
```

To view the configurations of a particular section in the Yum configuration file, run the following command:

```
yum-config-manager section…
```

You can also use a global regular expression to view the configurations of all matching sections.

```
yum-config-manager glob_expression…
```

For example, to view all configuration options and their values in the [main] section, run the following command:

```
$ yum-config-manager main \*
 Loaded plugins: langpacks, product-id, subscription-manager
 ================================== main ==================================
 [main]
 alwaysprompt = True
 assumeyes = False
 bandwith = 0
 bugtracker_url = https://bugzilla.redhat.com/enter_bug.cgi?product=Red%20Hat
%20Enterprise%20Linux%206&component=yum
 cache = 0
 [output truncated]
```

4.1.2 Creating a Yum Repository

To create a Yum repository, perform the following steps:

1. Run the following command as the root user to install the createrepo package:
   ```
   yum install createrepo
   ```
2. Copy all packages that you want to have in your repository into one directory, such as /mnt/local_repo/:
3. Change to this directory. Then, run the following command to create the necessary metadata for your Yum repository, as well as the sqlite database for speeding up Yum operations:
   ```
   createrepo --database /mnt/local_repo
   ```

4.1.3 Adding, Enabling, and Disabling a Yum Repository

This topic explains how to add, enable, and disable a repository by using the yum-config-manager command.

Adding a Yum Repository

To define a new repository, you can either add a [repository] section to the /etc/yum.conf file, or (recommended) add a .repo file in the //etc/yum.repos.d/ directory.

Yum repositories commonly provide their own .repo file. To add such a repository to your system, run the following command as the root user:

```
yum-config-manager --add-repo repository_url
```

For example, to add a repository located at http://www.example.com/example.repo, run the following command:

```
# yum-config-manager --add-repo http://www.example.com/example.repo
 Loaded plugins: langpacks, product-id, subscription-manager
 adding repo from: http://www.example.com/example.repo
 grabbing file http://www.example.com/example.repo to /etc/yum.repos.d/
example.repo
 example.repo                                               |  413 B     00:00
 repo saved to /etc/yum.repos.d/example.repo
```

Enabling a Yum Repository

To enable a particular repository or repositories, run the following command as the root user:

```
yum-config-manager --enable repository…
```

Alternatively, you can use a global regular expression to enable all matching Yum repositories:

```
yum-config-manager --enable glob_expression…
```

For example, to enable example, example-debuginfo, and example-source repositories, run the following command with a global regular expression:

```
# yum-config-manager --enable example\*
 Loaded plugins: langpacks, product-id, subscription-manager
 ============================== repo: example ===============================
 [example]
 bandwidth = 0
 base_persistdir = /var/lib/yum/repos/x86_64/6Server
 baseurl = http://www.example.com/repo/6Server/x86_64/
 cache = 0
 cachedir = /var/cache/yum/x86_64/6Server/example
 [output truncated]
```

Disabling a Yum Repository

To disable a particular repository or repositories, run the following command as the root user:

```
yum-config-manager --disable repository…
```

Alternatively, you can use a global regular expression to disable all matching Yum repositories:

```
yum-config-manager --disable glob_expression…
```

4.2 Managing Software Packages

Yum allows you to perform a complete set of operations with software packages, including searching for packages, viewing information about them, installing and removing.

Searching for Software Packages

To search for RPM packages by name, abbreviation, or description, run the following command:

```
yum search term…
```

For example, to search for the packages that match "meld or kompare", run the following command:

```
$ yum search meld kompare
 Loaded plugins: langpacks, langpacks, product-id, subscription-manager
 Updating Red Hat repositories.
```



```
INFO:rhsm-app.repolib:repos updated: 0
============================ N/S matched: kompare ============================
kompare.x86_64 : Diff tool
...
  Name and summary matches mostly, use "search all" for everything.
Warning: No matches found for: meld
```

Listing Software Packages

To list information on all installed and available RPM packages, run the following command:

```
yum list all
```

To list the installed and available RPM packages that match a particular global regular expression, run the following command:

```
yum list glob_expression…
```

For example, to list RPM packages with various ABRT add-ons and plug-ins, run the following command:

```
$ yum list abrt-addon\* abrt-plugin\*
 Loaded plugins: langpacks, product-id, subscription-manager
 Updating Red Hat repositories.
 INFO:rhsm-app.repolib:repos updated: 0
 Installed Packages
 abrt-addon-ccpp.x86_64                        1.0.7-5.el6             @rhel
 abrt-addon-kerneloops.x86_64                  1.0.7-5.el6             @rhel
 abrt-addon-python.x86_64                      1.0.7-5.el6             @rhel
 abrt-plugin-bugzilla.x86_64                   1.0.7-5.el6             @rhel
 abrt-plugin-logger.x86_64                     1.0.7-5.el6             @rhel
 abrt-plugin-sosreport.x86_64                  1.0.7-5.el6             @rhel
 abrt-plugin-ticketuploader.x86_64             1.0.7-5.el6             @rhel
```

Displaying Software Package Information

To display information about one or more RPM packages, run the following command:

```
yum info package_name…
```

For example, to display information about the abrt package, run the following command:

```
$ yum info httpd
Available Packages
Name        : httpd
Arch        : x86_64
Version     : 2.4.6
Release     : 40.4.h1
Size        : 1.2 M
Repo        : EulerOS-base
Summary     : Apache HTTP Server
URL         : http://httpd.apache.org/
License     : ASL 2.0
Description : The Apache HTTP Server is a powerful, efficient, and extensible
            : web server.
```

Installing Software Packages

To install a single package and all of its non-installed dependencies, run the following command as the root user:

```
yum install package_name
```

You can also install multiple packages simultaneously by appending their names as arguments. To do so, run the following command as the root user:

```
yum install package_name package_name…
```

For example, to install the sqlite package for the i686 architecture, run the following command:

```
# yum install sqlite.i686
```

Downloading Software Packages

At certain point of installation process, you are prompted to confirm the installation with the following message:

```
...
 Total size: 1.2 M
 Is this ok [y/d/N]:
 ...
```

If you select the d option, Yum will download the packages without installing them immediately. You can install these packages later in offline mode. By default, the downloaded packages are saved in the /var/cache/yum/$basearch/$releasever/packages/ directory.

Removing Software Packages

To uninstall a particular package and any packages that depend on it, run the following command as the root user:

```
yum remove package_name…
```

For example, to remove the totem package, run the following command:

```
# yum remove totem
```

4.3 Managing Software Package Groups

A package group is a collection of packages that serve a common purpose, for instance, System Tools. With Yum, you can perform an operation on a group of software packages simultaneously, saving time considerably.

Listing Software Package Groups

To view the number of installed groups, available groups, and available environment groups, run the following command with the summary option:

```
yum groups summary
```

Example command output:

```
$ yum groups summary
 Loaded plugins: langpacks, product-id, subscription-manager
 Available Environment Groups: 12
 Installed Groups: 10
 Available Groups: 12
```

To list all package groups and their group IDs, run the following command with the list ids option:

```
yum group list ids
```

To list a particular package group, for example, a package group related to the KDE desktop environment, run the following command:


```
$ yum group list ids kde\*
 Available environment groups:
  KDE Plasma Workspaces (kde-desktop-environment)
 Done
```

Displaying Software Package Group Information

To list mandatory and optional packages contained in a particular group, run the following command:

```
yum group info glob_expression…
```

For example, to display information about the LibreOffice package group, run the yum group info LibreOffice command.

```
$ yum group info LibreOffice
 Loaded plugins: langpacks, product-id, subscription-manager
 Group: LibreOffice
  Group-Id: libreoffice
  Description: LibreOffice Productivity Suite
  Mandatory Packages:
   =libreoffice-calc
  libreoffice-draw
   -libreoffice-emailmerge
  libreoffice-graphicfilter
   =libreoffice-impress
   =libreoffice-math
   =libreoffice-writer
   +libreoffice-xsltfilter
  Optional Packages:
  libreoffice-base
  libreoffice-pyuno
```

Installing a Software Package Group

Each software package group has a name and a groupid. You can install a package group by passing its group name or groupid to the group install command.

To install a software package group, run the following command as the root user:

```
yum group install "group name"
yum group install groupid
```

For example, to install a software package group related to the KDE desktop environment, run the following command:

```
# yum group install "KDE Desktop"
# yum group install kde-desktop
```

Removing a Software Package Group

To remove a software package group, run the yum group remove command with either its group name or groupid as the root user:

```
yum group remove group_name
yum group remove groupid
```

For example, to remove a software package group related to the KDE desktop environment, run the following command:

```
# yum group remove "KDE Desktop"
# yum group remove kde-desktop
```

4.4 Checking for and Updating Software Packages

Yum checks whether your system has any updates that wait to be applied. You can list the software packages that need to be updated and update them as a whole, or you can update a selected individual package.

Checking for Updates

To see which installed packages on your system have updates available, run the following command:

```
yum check-upd ate
```

Example command output:

```
# yum check-update
 Loaded plugins: langpacks, product-id, subscription-manager
 Updating Red Hat repositories.
 INFO:rhsm-app.repolib:repos updated: 0
 PackageKit.x86_64                0.5.8-2.el6                rhel
 PackageKit-glib.x86_64           0.5.8-2.el6                rhel
 PackageKit-yum.x86_64          0.5.8-2.el6                rhel
 PackageKit-yum-plugin.x86_64     0.5.8-2.el6                rhel
 glibc.x86_64                     2.11.90-20.el6             rhel
 glibc-common.x86_64              2.10.90-22                 rhel
 kernel.x86_64                    2.6.31-14.el6              rhel
 rpm.x86_64                       4.7.1-5.el6                rhel
 rpm-libs.x86_64                  4.7.1-5.el6                rhel
 rpm-python.x86_64                4.7.1-5.el6                rhel
 yum.noarch                       3.2.24-4.el6               rhel
```

Updating Software Packages

To update a single package, run the following command as the root user:

```
yum update package_name
```

For example, to update the rpm package, run the yum update rpm command.

```
# yum update rpm
 Loaded plugins: langpacks, product-id, subscription-manager
 Updating EulerOS repositories.
 INFO:rhsm-app.repolib:repos updated: 0
 Setting up Update Process
 Resolving Dependencies
 --> Running transaction check
 ---> Package rpm.x86_64 0:4.11.1-3.el7 will be updated
 --> Processing Dependency: rpm = 4.11.1-3.el7 for package: rpm-
libs-4.11.1-3.el7.x86_64
 --> Processing Dependency: rpm = 4.11.1-3.el7 for package: rpm-
python-4.11.1-3.el7.x86_64
 --> Processing Dependency: rpm = 4.11.1-3.el7 for package: rpm-
build-4.11.1-3.el7.x86_64
 ---> Package rpm.x86_64 0:4.11.2-2.el7 will be an update
 --> Running transaction check
 ...
 --> Finished Dependency Resolution
 Dependencies Resolved
 ================================================================================
  Package                   Arch        Version         Repository       Size
 ================================================================================
 Updating:
  rpm                       x86_64      4.11.2-2.el7    rhel            1.1 M
```



```
Updating for dependencies:
 rpm-build                x86_64      4.11.2-2.el7    rhel            139 k
 rpm-build-libs           x86_64      4.11.2-2.el7    rhel             98 k
 rpm-libs                 x86_64      4.11.2-2.el7    rhel            261 k
 rpm-python               x86_64      4.11.2-2.el7    rhel             74 k
Transaction Summary
================================================================================
Upgrade  1 Package (+4 Dependent packages)
Total size: 1.7 M
Is this ok [y/d/N]:
```

Similarly, it is possible to update a software package group by running the following command as the root user:

```
yum group update group_name
```

Updating All Software Packages and Their Dependencies

To update all software packages and their dependencies, run the following command as the root user:

```
yum update
```


5 Service Management

This topic describes how to manage your operating system and services using the systemd.

5.1 Introduction to systemd

The systemd is a system and service manager for Linux operating systems. It is designed to be backward compatible with SysV and LSB init scripts, and provides a number of features such as Socket & D-Bus based activation of services, on-demand activation of daemons, system state snapshots, and mount & automount point management. With systemd, the service control logic and parallelization are refined.

Systemd Units

In systemd, the targets of most actions are units, which are resources systemd know how to manage. Units are categorized by the type of resources they represent and defined in unit configuration files. For example, the avahi.service unit represents the Avahi daemon and is defined in the **avahi.service** file. **Table 5-1** lists available types of systemd units.

Table 5-1 Available types of systemd units

Unit Type	File Extension	Description
Service unit	.service	A system service.
Target unit	.target	A group of systemd units.
Automount unit	.automount	A file system automount point.
Device unit	.device	A device file recognized by the kernel.
Mount unit	.mount	A file system mount point.
Path unit	.path	A file or directory in a file system.
Scope unit	.scope	An externally created process.
Slice unit	.slice	A group of hierarchically organized units that manage system processes.
Snapshot unit	.snapshot	A saved state of the systemd manager.
Socket unit	.socket	An inter-process communication socket.
Swap unit	.swap	A swap device or a swap file.
Timer unit	.timer	A systemd timer.

All available types of systemd units are located in one of the following directories listed in **Table 5-2**.

Table 5-2 Locations of available systemd units

Directory	Description
/usr/lib/systemd/system/	Systemd units distributed with installed RPM packages.
/run/systemd/system/	Systemd units created at runtime.
/etc/systemd/system/	Systemd units created and managed by the system administrator.

5.2 Features

Fast Activation

The systemd provides more aggressive parallelization than UpStart. The use of Socket- and D-Bus based activation reduces the time required to boot the operating system.

To accelerate system boot, systemd seeks to:

- Activate only the necessary processes
- Activate as many processes as possible in parallel

UpStart are endeavoring to do the same thing with an event-triggered mechanism. A service is not started until a trigger event occurs and it can be activated together with irrelevant services.

On-Demand Activation

SysVinit is a type of init system designed prior to systemd. During initialization, SysVinit activates all the possible background service processes that might be used, although some of them, such as CUPS and SSHD, are rarely or even never used during system runtime. Users have to wait for login until all these service processes are activated. The drawbacks in SysVinit are obvious: slow system boot and a waste of system resources in activating unnecessary service processes.

Things turn around with systemd. A service can be activated on demand, and deactivated when it is no longer in use.

Service Life Cycle Management by CGroups

An important role of an init system is to track and manage the life cycle of services, and start/stop them freely. However, it is more difficult than you could ever imagine to encode an init system into stopping services freely.

Things are made simpler with CGroups, which has long been used to manage system resource quotas. The ease of use comes largely from its file-system-like user interface. When a parent service creates a child service, the latter inherits all attributes of the control group to which the parent service belongs. This means that all relevant services are put into the same control group. The systemd can find the PIDs of all relevant services simply by traversing their control group and then stop them one by one.

Mount and Automount Point Management

Traditional Linux systems use the /etc/fstab file to manage file system mount points. These mount points are automatically mounted at system boot time and usually are critical directories, such as the HOME directory. Like SysVinit, systemd also monitors and manages mount points so that they can be automatically mounted at system boot time. In systemd, you can continue to use the /etc/fstab file to manage mount points.

There are times when you need to mount or unmount on demand. This is traditionally achieved using the autofs service.

The systemd allows automatic mount without a need to install autofs.

Transactional Dependency Management

System boot involves a host of separate jobs, some of which may be dependent on each other. For example, a network file system (NFS) can be mounted only after network connectivity is activated. The systemd can run a large number of dependent jobs in parallel, but not all of them. Looking back to the NFS example, it is impossible to mount NFS and activate network at the same time. Before running a job, systemd calculates its dependencies, creates a temporary transaction, and verifies that this transaction is consistent (all relevant services can be activated without any dependency on each other).

Compatibility with SysV Init Scripts

The systemd differs from its predecessor in configuration mode and application development requirements, but is still compatible with its predecessor considering that no Linux distribution would be willing to replace the current init system with systemd at the cost of overhauling its service code.

The compatibility between systemd and SysV and LSB init scripts allows you to upgrade your system to systemd without any changes to existing services or processes, making it easier for systemd to be widely accepted by users.

System State Snapshots and System Restoration

System state varies constantly because services can be activated on demand at any given point in time. The systemd can temporarily save the current state of your operating system or restore a previous state of the operating system from a dynamically created snapshot.

Imagine a system snapshot is created while services A and B are running, and then service A is stopped and some system changes (such as activation of service C) are applied. With the system snapshot, you can return your operating system to the state in which services A and B were running. Snapshot-based restoration is very helpful in debugging scenarios — you can undo any debugging operations when the debug is completed.

5.3 Managing System Services

The systemd provides the systemctl command to start, stop, restart, view, enable, and disable system services.

Comparison Between SysVinit and systemd Commands

The service and chkconfig commands from SysVinit serve similar purpose as the systemctl command from systemd. Note that service and chkconfig commands are included in systemd only for compatibility reasons and should be avoided.

Table 5-3 Comparison between SysVinit and systemd commands

SysVinit Command	systemd Command	Description
service foo start	systemctl start foo.service	Starts a service.
service foo stop	systemctl stop foo.service	Stops a service.
service foo restart	systemctl restart foo.service	Restarts a service.

SysVinit Command	systemd Command	Description
service foo reload	systemctl reload foo.service	Reloads a configuration file without interrupting an operation.
service foo condrestart	systemctl condrestart foo.service	Restarts a service only if it is running.
service foo status	systemctl status foo.service	Checks if a service is running.
chkconfig foo on	systemctl enable foo.service	Enables a service when the service activation time arrives or a trigger condition for enabling the service is met.
chkconfig foo off	systemctl disable foo.service	Disables a service when the service activation time arrives or a trigger condition for disabling the service is met.
chkconfig foo	systemctl is-enabled foo.service	Checks whether a service is enabled.
chkconfig –list	systemctl list-unit-files --type=service	Lists all services in each runlevel and checks whether they are enabled.
chkconfig foo –list	ls /etc/systemd/system/*.wants/foo.service	Lists the runlevels in which a service is enabled and those in which the service is disabled.
chkconfig foo –add	systemctl daemon-reload	Reloads the systemd process. Used when you need to create a service file or change settings.

Listing Services

To list all currently loaded services, run the following command:

```
systemctl list-units --type service
```

To list all services regardless of whether they are loaded, run the following command (with the all option):

```
systemctl list-units --type service --all
```

Example list of all currently loaded services:

```
$ systemctl list-units --type service
```



```
 UNIT                            LOAD ACTIVE SUB     DESCRIPTION
 abrt-ccpp.service               loaded active exited  Install ABRT coredump hook
 abrt-oops.service               loaded active running ABRT kernel log watcher
 abrt-vmcore.service             loaded active exited  Harvest vmcores for ABRT
 abrt-xorg.service               loaded active running ABRT Xorg log watcher
 abrtd.service                   loaded active running ABRT Automated Bug
Reporting Tool
 ...
 systemd-vconsole-setup.service loaded active exited  Setup Virtual Console
 tog-pegasus.service             loaded active running OpenPegasus CIM Server
 LOAD = Reflects whether the unit definition was properly loaded.
 ACTIVE = The high-level unit activation state, i.e. generalization of SUB.
 SUB    = The low-level unit activation state, values depend on unit type.
 46 loaded units listed. Pass --all to see loaded but inactive units, too.
 To show all installed unit files use 'systemctl list-unit-files'
```

Displaying Service Status

To display the status of a service, run the following command:

```
systemctl status name.service
```

Table 5-4 describes the parameters in the command output.

Table 5-4 Output parameters

Parameter	Description
Loaded	Information on whether the service has been loaded, the absolute path to the service file, and a note of whether the service is enabled.
Active	Information on whether the service is running and a time stamp.
Main PID	PID of the service.
Status	Additional information about the service.
Process	Additional information about related processes.
CGroup	Additional information about related control groups.

To verify whether a particular service is running, run the following command:

```
systemctl is-active name.service
```

Similarly, to determine whether a particular service is enabled, run the following command:

```
systemctl is-enabled name.service
```

For example, to display the status of gdm.service, run the systemctl status gdm.service command.

```
# systemctl status gdm.service
 gdm.service - GNOME Display Manager
  Loaded: loaded (/usr/lib/systemd/system/gdm.service; enabled)
  Active: active (running) since Thu 2013-10-17 17:31:23 CEST; 5min ago
  Main PID: 1029 (gdm)
```



```
  CGroup: /system.slice/gdm.service
            ├─1029 /usr/sbin/gdm
            ├─1037 /usr/libexec/gdm-simple-slave --display-id /org/gno...
            └─1047 /usr/bin/Xorg :0 -background none -verbose -auth /r...
 Oct 17 17:31:23 localhost systemd[1]: Started GNOME Display Manager.
```

Starting a Service

To start a service, run the following command as the root user:

```
systemctl start name.service
```

For example, to start the httpd service, run the following command:

```
# systemctl start httpd.service
```

Stopping a Service

To stop a service, run the following command as the root user:

```
systemctl stop name.service
```

For example, to stop the bluetooth service, run the following command:

```
# systemctl stop bluetooth.service
```

Restart a Service

To restart a service, run the following command as the root user:

```
systemctl restart name.service
```

This command stops the selected service in the current session and immediately starts it again. If the selected service is not running, this command starts it too.

For example, to restart the bluetooth service, run the following command:

```
# systemctl restart bluetooth.service
```

Enabling a Service

To configure a service to start automatically at system boot time, run the following command as the root user:

```
systemctl enable name.service
```

For example, to configure the httpd service to start automatically at system boot time, run the following command:

```
# systemctl enable httpd.service
 ln -s '/usr/lib/systemd/system/httpd.service' '/etc/systemd/system/multi-
user.target.wants/httpd.service'
```

Disabling a Service

To prevent a service from starting automatically at system boot time, run the following command as the root user:

```
systemctl disable name.service
```

For example, to prevent the bluetooth service from starting automatically at system boot time, run the following command:

```
# systemctl disable bluetooth.service
 rm '/etc/systemd/system/dbus-org.bluez.service'
 rm '/etc/systemd/system/bluetooth.target.wants/bluetooth.service'
```

5.4 Changing a Runlevel

Targets and Runlevels

In systemd, the concept of runlevels has been replaced with systemd targets to improve flexibility. For example, you can inherit an existing target and turn it into your own target by adding other services. The table below provides a complete list of runlevels and their corresponding systemd targets.

Table 5-5 Mapping between runlevels and targets

Runlevel	systemd Target	Description
0	runlevel0.target, poweroff.target	The operating system is powered off.
1	runlevel1.target, rescue.target	The operating system is in single user mode.
2	runlevel2.target, multi-user.target	The operating system is in user-defined or domain-specific runlevel (by default, it is equivalent to runlevel 3).
3	runlevel3.target, multi-user.target	The operating system is in non-graphical multi-user mode, and can be accessed from multiple consoles or networks.
4	runlevel4.target, multi-user.target	The operating system is in user-defined or domain-specific runlevel (by default, it is equivalent to runlevel 3).
5	runlevel5.target, graphical.target	The operating system is in graphical multi-user mode.
6	runlevel6.target, reboot.target	The operating system is rebooted.

Viewing the Default Target

To determine which target is used by default, run the following command:

```
systemctl get-default
```

Example command output:

```
$ systemctl get-default
 graphical.target
```

Viewing the Current Target

To list all currently loaded targets, run the following command:

```
systemctl list-units --type target
```

Example command output:

```
$ systemctl list-units --type target
 UNIT                   LOAD ACTIVE SUB    DESCRIPTION
 basic.target           loaded active active Basic System
 cryptsetup.target      loaded active active Encrypted Volumes
 getty.target           loaded active active Login Prompts
 graphical.target       loaded active active Graphical Interface
 local-fs-pre.target loaded active active Local File Systems (Pre)
 local-fs.target        loaded active active Local File Systems
 multi-user.target      loaded active active Multi-User System
 network.target         loaded active active Network
 paths.target           loaded active active Paths
 remote-fs.target       loaded active active Remote File Systems
 sockets.target         loaded active active Sockets
 sound.target           loaded active active Sound Card
 spice-vdagentd.target loaded active active Agent daemon for Spice guests
 swap.target            loaded active active Swap
 sysinit.target         loaded active active System Initialization
 time-sync.target       loaded active active System Time Synchronized
 timers.target          loaded active active Timers
 LOAD = Reflects whether the unit definition was properly loaded.
 ACTIVE = The high-level unit activation state, i.e. generalization of SUB.
 SUB    = The low-level unit activation state, values depend on unit type.
 17 loaded units listed. Pass --all to see loaded but inactive units, too.
 To show all installed unit files use 'systemctl list-unit-files'.
```

Changing the Default Target

To change the default target, run the following command as the root user:

```
systemctl set-default name.target
```

Example command output:

```
# systemctl set-default multi-user.target
 rm '/etc/systemd/system/default.target'
 ln -s '/usr/lib/systemd/system/multi-user.target' '/etc/systemd/system/
default.target'
```

Changing the Current Target

To change the current target, run the following command as the root user:

```
ssystemctl isolate name.target
```

Example command output:

```
# systemctl isolate multi-user.target
```

Changing to Rescue Mode

To change the operating system to rescue mode, run the following command as the root user:

```
systemctl rescue
```


This command is similar to systemctl isolate rescue.target, but it also sends an informative message to all login users. To prevent systemd from sending this message, run this command with the --no-wall option:

```
systemctl --no-wall rescue
```

For example:

```
# systemctl rescue
 Broadcast message from root@localhost on pts/0 (Fri 2013-10-25 18:23:15 CEST):
 The system is going down to rescue mode NOW!
```

Changing to Emergency Mode

To change the operating system to emergency mode, run the following command as the root user:

```
systemctl emergency
```

This command is similar to systemctl isolate emergency.target, but it also sends an informative message to all login users. To prevent systemd from sending this message, run this command with the --no-wall option:

```
systemctl --no-wall emergency
```

5.5 Shutting Down, Restarting, Suspending, and Hibernating the Operating System

systemctl Command

The systemd uses the systemctl command instead of old Linux system management commands to shut down, restart, suspend, and hibernate the operating system. Although old Linux system management commands are still available in systemd for compatibility reasons, but it is advised that you use systemctl when possible.

Table 5-6 Mapping between old Linux system management commands and systemctl

Old Command	systemctl Command	Description
halt	systemctl halt	Shuts down the operating system.
poweroff	systemctl poweroff	Powers off the operating system.
reboot	systemctl reboot	Reboots the operating system.
pm-suspend	systemctl suspend	Suspends the operating system.
pm-hibernate	systemctl hibernate	Hibernates the operating system.
pm-suspend-hybrid	systemctl hybrid-sleep	Hibernates and suspends the operating system.

Shutting Down the Operating System

To shut down the system and power off the operating system, run the following command as the root user:

```
systemctl poweroff
```

To shut down the operating system without powering it off, run the following command as the root user:

```
systemctl halt
```

By default, running either of these commands causes systemd to send an informative message to all login users. To prevent systemd from sending this message, run the selected command with the --no-wall option:

```
systemctl --no-wall poweroff
```

Restarting the Operating System

To restart the operating system, run the following command as the root user:

```
systemctl reboot
```

Running this command causes systemd to send an informative message to all login users. To prevent systemd from sending this message, run this command with the --no-wall option:

```
systemctl --no-wall reboot
```

Suspending the Operating System

To suspend the operating system, run the following command as the root user:

```
systemctl suspend
```

Hibernating the Operating System

To hibernate the operating system, run the following command as the root user:

```
systemctl hibernate
```

To suspend and hibernate the operating system, run the following command as the root user:

```
systemctl hybrid-sleep
```


6 Process Management

This topic explains how Linux kernel manages processes. It also provides examples to help you better understand common process control commands, at and cron services, as well as process query commands.

6.1 Managing System Processes

In most cases, the operating system comes with only one CPU and one main memory, but it may have multiple tier-2 disks and input/output (I/O) devices. It is impossible for the operating system to take care of only one user at a given point in time. Therefore, users have to share resources, but it appears to users that they are exclusively occupying resources. This is achieved thanks to the use of a task queue. The operating system places user tasks, OS tasks, emailing, print tasks, and other pending tasks in the queue and schedules the tasks according to predefined rules. In this topic, you will know how the operating system manages processes.

6.1.1 Starting a Process Manually

6.1.2 Scheduling a Process

The time-consuming and resource-demanding part of maintenance work is often performed at late night. You can arrange relevant processes to get started at the scheduled time instead of staying up all night. Here, we will explain the process scheduling commands.

6.1.2.1 Using the at Command to Run Processes at the Scheduled Time

Function

The at command is used to run a batch of processes (a series of commands) at the scheduled time or time+date.

Syntax of the at command:

```
at [-V] [-q queue] [-f filename] [-mldbv] time
 at -cjob[job…]
```

Time Format

The scheduled time can be in any of the following formats:

- hh:mm today: If hh:mm is earlier than the current time, the selected commands will be run at hh:mm the next day.
- midnight, noon, teatime (typically at 16:00), or the like
- 12-hour format followed by am or pm
- Time + date (month day, mm/dd/yy, or dd.mm.yy)

The scheduled time can also be relative time. For example, now+*N* minutes, hours, days, or weeks. Further, the scheduled time can be words like today, tomorrow, or the like.

Imagine the current time is 12:30 June 7 2015 and you want to run a command at 4:30 pm. The scheduled time in the at command can be any of the following:

```
at 4:30pm
 at 16:30
 at 16:30 today
 at now+4 hours
 at now+ 240 minutes
 at 16:30 7.6.15
 at 16:30 6/7/15
 at 16:30 Jun 7
```

Absolute time in 24-hour format, such as at 16:30 6/7/15, is recommended.

Privileges

Only commands from standard input or from the file specified by the -f option can be scheduled by the at command to be executed. If the su command is executed to switch the operating system from user A to user B and then the at command is executed at the shell prompt of user B, the at command execution result is sent to user B.

For example, to run the slocate -u command at 10 am on 8 June 2015, perform the following steps:

1. Type **at 10 : 00 6/8/15** at the shell prompt.
2. When the at> prompt appears, type **slocate -u** and press Enter.
3. Repeat substep 2 to add other commands that need to be run at 10 am on 8 June 2015. Then, press Ctrl+d to exit the at command.

```
# at  10 : 00  6/8/15
 warning:commands will be executed using (in order) a) $SHELL b) login shell
c) /bin/sh
 at> slocate -u
 at>
```



```
[1]+ Stopped    at  10:00  6/8/15
```

The administrator is authorized to run the at command unconditionally. For other users, their privilege to run the at command is defined in /etc/at.allow and /etc/at.deny files.

6.1.2.2 Using the cron Service to Run Commands Periodically

The at command can run commands at the scheduled time but only once. If you need to run commands repeatedly, the cron service is a good helper.

Cron Service

Each user has a crontab file. This file contains periodic commands and the command execution interval. The file name is the same as the user name, which is defined in the /etc/passwd file. For example, the crontab file of the globus user is /var/spool/cron/globus. The cron service searches the /var/spool/cron directory every minute for crontab files and loads the search results into memory to execute the commands in the crontab files. Command execution results are then mailed to users specified by the environment variable MAILTO in the /etc/crontab file.

If no crontab files are found, the cron service enters sleep mode and releases system resources. One minute later, the cron service is awaken to repeat the search work and command execution.

The cron service also reads the cron configuration file /etc/crontab every minute.

The cron service, once started, does not require manual intervention except when you need to replace periodic commands with new ones.

crontab Command

The crontab command is used to install, edit, remove, list, and perform other operations on crontab files.

Here are common crontab command options:

- crontab -u //Set the cron service of a user. This option is required only when the crontab command is run by the root user.
- crontab -l //List details of the cron service of a user.
- crontab -r //Remove the cron service of a user.
- crontab -e //Edit the cron service of a user.

For example, to list cron service settings of the root user, run the following command:

```
crontab -u root -l
```

crontab Files

Format of each line in a crontab file:

```
minute hour day-of-month month-of-year day-of-week commands
```

Each line in a crontab file consists of 6 fields separated by space or space tab.

Table 6-1 Field description

Field	Description
minute	The minute of the hour at which commands will be executed. Value range: 0 – 59.
hour	The hour of the day at which periodic commands will be executed. Value range: 0 – 23.
day-of-month	The day of month at which periodic commands will be executed. Value range: 1 – 31.
month-of-year	The month of year at which periodic commands will be executed. Value range: 1 – 12.
day-of-week	The day of week at which periodic commands will be executed. Value range: 0 – 6.
commands	Periodic commands.

The fields cannot be left unspecified. In addition to numerical values, the following special symbols are allowed:

- Asterisk (*): a wildcard value.
- Forward slash (/): followed by a numeral N to indicate that commands will be executed at a regular interval of N.
- Hyphen (-): used with a range.
- Comma (,): used to separate discrete numbers.

For example, to allow the operating system to add sleepy to the /tmp/test.txt file every two hours from 11 pm to 08 am, add the following line in a crontab file:

```
* 23-8/2 * * * echo"sleepy" >> /tmp/test.txt
```

Each time the cron service settings of a user are edited, the cron service generates in the /var/spool/cron directory a crontab file named after the user. The crontab file can be edited only using the crontab -e command. Alternatively, the user can create a new file and run the crontab *filename* command to import its cron settings into the new file.

For example, to create a crontab file for the globus user, perform the following steps:

1. Create a new file using any text editor. Add the commands that need to be executed periodically and the command execution interval to the new file. In this example, the new file is ~/globus.cron.
2. Run the following command to install the new file as the crontab file of the globus user:
   ```
   crontab  globus. -/globus.cron
   ```

After the new file is installed, you will find a file named globus in the /var/spool/cron directory.

NOTE

Do not restart the cron service after a crontab file is modified, because the cron service, once started, reads the crontab file every minute to check whether there are commands that need to be executed periodically.

/etc/crontab File

A crontab file contains user-specific commands, whereas the /etc/crontab file contains system-wide commands. By reading the /etc/crontab file every minute, the cron service finds whether there are system-wide commands waiting to be executed periodically. An absolute path to the commands shall be provided. Example /etc/crontab file

```
SHELL=/bin/sh
 PATH=/usr/bin:/usr/sbin:/sbin:/bin:/usr/lib/news/bin
 MAILTO=root  //Send an email to the root user if a command output is generated
or an error occurs in executing the commands
 HOME=/
 #  run-parts
 01  * * * * root run-parts /etc/cron.hourly     //Execute the scripts in /etc/
cron.hourly once every hour
 02 4 * * * root run-parts /etc/cron.daily    //Execute the scripts in /etc/
cron.daily once every day
 22 4 * * 0  root run-parts /etc/cron.weekly     //Execute the scripts in /etc/
cron.weekly once every week
 42 4 1  * *  root run-parts /etc/cron.monthly     //Execute the scripts in /etc/
cron.monthly once every month
```

NOTE

Periodic commands can be listed in either of the following formats: (1) command names; (2) run-parts *directory where periodic commands are located*.

6.1.3 Suspending/Resuming a Process

To suspend a foreground process, press Ctrl+Z. After you press Ctrl+Z, the cat command is suspended together with the foreground process you wish to suspend. You can use the jobs command instead to display a list of shell jobs, including their job names, IDs, and status.

To resume a process in foreground or background, run the fg or bg command, respectively. The process then starts from where it paused previously.

6.2 Viewing Processes

Linux is a multi-task system and needs to get process information during process management. Multiple commands are available to view processes.

who Command

The who command is used to display system user information. For example, before running the talk command to establish instant communication with another user, you need to run the who command to determine whether the target user is online. As another example, the system administrator can run the who command to learn what each login user is doing at the current time. The who command is widely seen in system administration since it is easy to use and can return a comprehensive set of accurate user information.

The following is an example output of the who command, where system users and their status are displayed:


```
# who
 admin     tty1         Jul 28 15:55
 admin     pts/0        Aug  5 15:46 (9.1.0.110)
 admin     pts/2        Jul 29 19:52 (9.1.0.110)
 root     pts/3        Jul 30 12:07 (9.1.0.110)
 root     pts/4        Jul 31 10:29 (9.1.0.144)
 root     pts/5        Jul 31 14:52 (9.1.0.11)
 root     pts/6        Aug  6 10:12 (9.1.0.234)
 root     pts/8        Aug  6 11:34 (9.1.0.234)
```

ps Command

The ps command is used to display process information, including which processes are running, terminated, resource-hungry, or stay as zombies.

A common scenario is using the ps command to monitor background processes, which do not interact with your screen, keyboard, and other I/O devices. **Table 6-2** lists the common ps command options.

Table 6-2 Common ps command options

Option	Description
-e	Displays all processes.
-f	Full output format.
-h	Hides column headings in the listing of process information.
-l	Long output format.
-w	Wide output format.
-a	Lists all processes on a terminal, including those of other users.
-r	Lists only running processes.
-x	Lists all processes without controlling terminals.

For example, to list all processes on a terminal, run the following command:

```
# ps -a
   PID TTY          TIME CMD
 12175 pts/6    00:00:00 bash
 24526 pts/0    00:00:00 vsftpd
 29478 pts/5    00:00:00 ps
 32461 pts/0    1-01:58:33 sh
```

top Command

Both the top and the ps commands can display a list of currently running processes, but the top command allows you to update the displayed list of processes repeatedly with the press of a button. If the top command is executed in foreground, it exclusively occupies foreground until it is terminated.

The top command provides real-time visibility into system processor status. You can sort the list of CPU tasks by CPU usage, memory usage, or task execution time. Extensive customization of the display, such as choice of columns or sorting method, can be achieved using interactive commands or the customization file.

Figure 6-1 provides an example output of the top command.

Figure 6-1 Example command output

```
top - 19:04:08 up 9 days,  3:09,  8 users,  load average: 2.17, 2.08, 2.06
Tasks: 242 total,   8 running, 234 sleeping,   0 stopped,   0 zombie
Cpu(s):  8.3%us,  0.2%sy,  0.0%ni, 91.5%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:     19983M total,    19777M used,      206M free,      567M buffers
Swap:     2053M total,       10M used,     2043M free,    12326M cached

  PID USER      PR  NI  VIRT  RES  SHR S  %CPU %MEM    TIME+  COMMAND
32757 root      20   0 4462m 3.0g 5440 S   100 15.3   1542:40 qemu-kvm
32461 root      20   0 11580 1380 1120 R   100  0.0   1563:47 sh
31437 root      20   0 4626m 2.4g 5436 R     4 12.1  14:36.89 qemu-kvm
29553 root      20   0 17256 1392  932 R     0  0.0   0:00.02 top
31438 root      20   0     0    0    0 S     0  0.0   0:12.80 vhost-31437
32758 root      20   0     0    0    0 S     0  0.0   0:25.21 vhost-32757
    1 root      20   0 10540  796  748 S     0  0.0   0:04.59 init
    2 root      20   0     0    0    0 S     0  0.0   0:00.00 kthreadd
    3 root      20   0     0    0    0 S     0  0.0   0:01.64 ksoftirqd/0
    6 root      RT   0     0    0    0 S     0  0.0   0:01.08 migration/0
    7 root      RT   0     0    0    0 S     0  0.0   0:01.66 watchdog/0
    8 root      RT   0     0    0    0 S     0  0.0   0:01.09 migration/1
    9 root      20   0     0    0    0 S     0  0.0   0:05.58 kworker/1:0
   10 root      20   0     0    0    0 S     0  0.0   0:01.31 ksoftirqd/1
   11 root      20   0     0    0    0 S     0  0.0   0:50.48 kworker/0:1
   12 root      RT   0     0    0    0 S     0  0.0   0:01.27 watchdog/1
   13 root      RT   0     0    0    0 S     0  0.0   0:01.64 migration/2
   14 root      20   0     0    0    0 S     0  0.0   0:00.00 kworker/2:0
   15 root      20   0     0    0    0 S     0  0.0   1:01.89 ksoftirqd/2
   16 root      RT   0     0    0    0 S     0  0.0   0:01.38 watchdog/2
   17 root      RT   0     0    0    0 R     0  0.0   0:01.12 migration/3
   18 root      20   0     0    0    0 S     0  0.0   0:00.00 kworker/3:0
   19 root      20   0     0    0    0 S     0  0.0   0:22.84 ksoftirqd/3
   20 root      RT   0     0    0    0 S     0  0.0   0:01.33 watchdog/3
   21 root      RT   0     0    0    0 S     0  0.0   0:01.56 migration/4
   22 root      20   0     0    0    0 S     0  0.0   0:00.00 kworker/4:0
   23 root      20   0     0    0    0 S     0  0.0   0:00.01 ksoftirqd/4
   24 root      RT   0     0    0    0 S     0  0.0   0:01.29 watchdog/4
```

kill Command

The kill command is used to terminate a process regardless of whether the process is running in foreground or background. It differs from the combo key Ctrl+c, which can terminate only foreground processes. The reason for terminating a background process can be heavy use of CPU resources or deadlock.

The kill command sends a signal to terminate running processes. By default, the TERM signal is used. The TERM signal terminates all processes incapable of capturing the TERM signal. To terminate a process capable of capturing the TERM signal, use the KILL signal (signal ID: 9) instead.

Two types of syntax of the kill command:


```
kill [-s signal | -p] [-a] PID…
 kill -l [signal]
```

The process ID is retrieved from the ps command. The -s option indicates the signal sent to terminate processes. The -p option indicates the ID of process that will be terminated.

For example, to terminate the process with ID 1409, run the following command:

```
# kill -9 1409
```

Example output of the kill command with the -l option

```
# kill -l
  1) SIGHUP? 2) SIGINT? 3) SIGQUIT? 4) SIGILL
  5) SIGTRAP? 6) SIGABRT? 7) SIGBUS? 8) SIGFPE
  9) SIGKILL?10) SIGUSR1?11) SIGSEGV?12) SIGUSR2
 13) SIGPIPE?14) SIGALRM?15) SIGTERM?16) SIGSTKFLT
 17) SIGCHLD?18) SIGCONT?19) SIGSTOP?20) SIGTSTP
 21) SIGTTIN?22) SIGTTOU?23) SIGURG?24) SIGXCPU
 25) SIGXFSZ?26) SIGVTALRM?27) SIGPROF?28) SIGWINCH
 29) SIGIO?30) SIGPWR?31) SIGSYS?34) SIGRTMIN
 35) SIGRTMIN+1?36) SIGRTMIN+2?37) SIGRTMIN+3?38) SIGRTMIN+4
 39) SIGRTMIN+5?40) SIGRTMIN+6?41) SIGRTMIN+7?42) SIGRTMIN+8
 43) SIGRTMIN+9?44) SIGRTMIN+10?45) SIGRTMIN+11?46) SIGRTMIN+12
 47) SIGRTMIN+13?48) SIGRTMIN+14?49) SIGRTMIN+15?50) SIGRTMAX-14
 51) SIGRTMAX-13?52) SIGRTMAX-12?53) SIGRTMAX-11?54) SIGRTMAX-10
 55) SIGRTMAX-9?56) SIGRTMAX-8?57) SIGRTMAX-7?58) SIGRTMAX-6
 59) SIGRTMAX-5?60) SIGRTMAX-4?61) SIGRTMAX-3?62) SIGRTMAX-2
```


7 kbox

7.1 Overview

This topic describes basic concepts and structure of the kernel box (kbox) and certain software and hardware requirements and restrictions.

7.1.1 Overview

This topic describes background and basic concepts of the kbox, to help you better understand the kbox.

Overview

The kernel of the Linux system is relative complex. Modules are closely associated with each other. Due to less efficient maintenance tools, the maintenance work is difficult to implement. Although the kernel is equipped with log recording systems such as the klogd and syslogd, logs may not be recorded (or cannot be recorded in a timely manner) in case of emergencies such as unexpected system reboot, kernel panic, and memory overflow. As a result, you cannot identify root causes of these problems.

To solve these problems and rescue lost kernel logs, the EulerOS provides the kbox feature. The kbox functions as the black box designed for the aircraft systems. Upon system exceptions, important information is recorded in a special channel (using the non-volatile storage). Therefore, you can analyze the system status upon the exceptions.

Feature Introduction

The kbox provides a mechanism that can record important information about the kernel upon system exceptions, and record the important information in the non-volatile storage devices. Based on the recorded information, you can analyze the causes of the system exceptions and locate the faults.

For example, when the kernel panic occurs, the kbox collects exception information generated by the kernel and saves the information in a temporary region. After collecting all information, the kbox dumps the information saved in the temporary region to the non-volatile storage devices (for example, the NVRAM) or the specified memory using the kdump function for further analysis.

NOTE

The kbox only collects the kernel output information and function calling relationships that result in the system exceptions instead of sensitive data of the users. For details about how to disable the kbox, see **7.6.1 Command References**.

Components of the kbox

Table 7-1 describes the components of the kbox module.

Table 7-1 Elements of the kbox module

Component	Description
Kbox	Manage the non-volatile storage devices, and capture and save information generated for exceptions.

Component	Description
Drive of the non-volatile storage device	Provide read and write interfaces for the kbox. When the kbox is enabled, the corresponding drive module is loaded to register the storage device with the kbox.
Non-volatile storage device	Save information generated for exceptions. When exceptions occur, the kbox dumps the exception information to the non-volatile storage device. If no non-volatile storage device is configured, the kbox saves the logs using the kdump function.

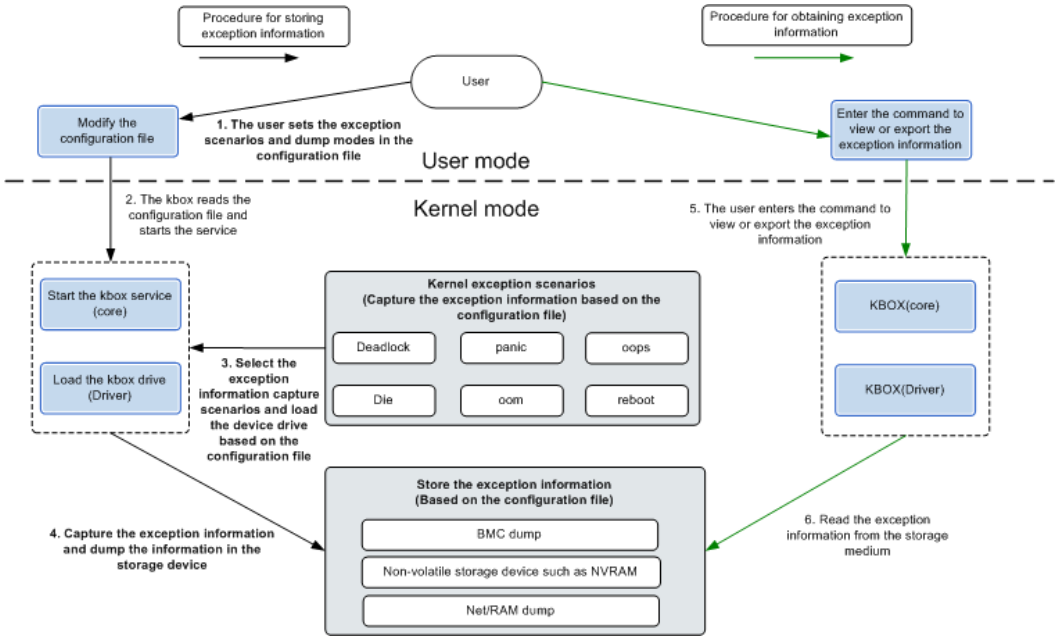
7.1.2 Kbox Structure

This topic describes the structure and service flow of the kbox.

Structure

The following figure shows the integrated structure and service flow of the kbox.

Figure 7-1 Kbox structure



NOTE

1. Temporarily, the exception information storage does not support the BMC or net/ram dump mode.

2. The current version supports only the following exception scenarios: OOM, OOPS (die), panic, and deadlock.

Service Flow

1. The kbox module modifies the configuration file. Based on the actual situations, the kbox module configures the product information, scenarios for capturing exception information, and dump devices based on the configuration file (**/etc/kbox/config**).
2. The kbox starts in service mode. After reading the configuration file (**/etc/kbox/config**), the kbox sends the product information and exception scenarios to the kbox module as module parameters.
3. Based on the storage device parameters specified in the configuration file (**/etc/kbox/config**), the kbox loads the drive of the corresponding storage device and registers the storage device with the kbox module.
4. When an exception occurs and the exception information is captured by the kbox module, the kbox module records the exception information in the temporary storage region, and then synchronizes the exception information to the storage device.
5. The user with the root rights can run the log export command to export the exception information to the **/var/log/kbox/** directory from the storage device.
6. By reading the exception information file, the user can obtain the exception information to identify the problems.

7.1.3 Software and Hardware Requirements

This topic describes the software and hardware requirements of the kbox.

- Software requirements

  The kbox supports only the EulerOS system.
- Hardware requirements

  The CloudEdge product uses normal memories and has no special hardware requirements.

7.2 Functions

This topic describes the exception scenarios, captured information, and parsing of the captured information.

7.2.1 Restrictions

This topic describes restricts of the kbox.

NOTE

It is prohibited to export sensitive information using the kbox.

- When exceptions are triggered, the kbox records only the earliest exception and dumps the exception information to the storage device. For example, when the panic, OOPS, or OOM exception occurs and then the same type of exceptions also occur, the kbox does not record the latter exception information but outputs prompt information in the system logs
- During information recording, if the CPU sends an interrupt to the kbox, the ongoing process is interrupted; in such a case, if another exception is triggered, the kbox only outputs prompt information to the system logs instead of recording logs (including the previous and latter exceptions) in the storage device.
- If the panic, OOPS, or OOM exception occurs when the exception information dumped by the kbox is exported using the **os_kbox_config** command line, there is a possibility that the dumped information is incomplete due to the concurrent mutex obtaining.
- When the stack is overflown downwards, the information obtained by the stack is all 0s. When the stack capacity is 8 KB, the information obtained by the stack is null.

7.2.2 Providing System Panic Information

As explained in this topic, the kbox can directly or indirectly call the panic function to record exception information when the system encounters an exception.

Function Introduction

Due to special exceptions, the product or platform service software may cause critical errors in the OS kernel. The kbox can record the exception information such as the time of occurrence and calling track, in the storage device, and therefore facilitating fault locating.

Information List

The recorded panic information includes three parts.

1. Panic information. Up to 32 KB information can be recorded. This area mainly records the following information:
   - Time of occurrence (UTC time)
   - Cause of the kernel panic
   - PID, TGID, and name of the exception process
   - Track of the called kernel functions and stack information (up to 150 lines of the stack information can be output)
   - VMA (virtual memory block of the process) information
   - Command line information of the current process
   - Environment variables of the system
   - Register information
2. Message output logs of the kernel. Up to 64 KB information can be recorded. This area mainly records the following information:
   - Time of occurrence (UTC time)
   - 64 KB of latest logs generated when the exception occurs, that is, the last 64 KB logs output by the kernel to the circular buffer

3. Message output logs of the console. Up to 32 KB information can be recorded. This area mainly records the following information:

- Time of occurrence (UTC time)
- Information about the kernel stack
- Track of called kernel functions
- Message output logs of other kernels

Examples

A log file includes the following information:

1. Information about the panic exception

```
//Panic exception storage area, the current panic log is located in the panic
exception storage area with index 0 *****area type:panic - location in panic
area:0***** ----------KBOX_START---------- //Time when kernel panic occurs.
The time is in UTC format. Beijing time = UTC time + 8 hours; panic time:
20080120231731-a7634 //panic reason:Watchdog detected hard LOCKUP on cpu 1
panic stack: //PID, TGID, and name of the exception process: <pid:
7344:7344:insmod> <kernel_stack> Stack: ffff880130467ea0 ffffffffa0015177
ffffffffa00e5740 ffffffffa00d7cb0 00000000000005a0 ffffffffa00e5b98
ffff880130467ef0 ffffffffa00d3aa4 ffff880130467ec0 ffffffffa00d7c60
ffff880130467ef0 ffffffffa00e5740 ffffffffa00e5760 ffff880106768000
431bde82d7b634db ffff880106768000 ffff880130467f40 ffffffffa00d3e1c
ffff880c4ee54700 0000000000000000 ffff8801064bd000 0000000000000000
0000000000000000 0000000000000000 0000000000000000 0000000000000000
0000000000000000 ffffffff8145a5d4 //Call Trace: <NMI>  [<ffffffffa0002220>]
kbox_show_task_kernel_info+0x200/0x300 [kbox] [<ffffffffa00054c7>]
kbox_print_specified_tasks+0x17/0x50 [kbox] [<ffffffffa0000174>]
kbox_panic_notifier_callback+0x174/0x210 [kbox] [<ffffffff811f7e4f>] ?
__const_udelay+0x2f/0x40 [<ffffffff8102156f>] ? native_safe_apic_wait_icr_idle
+0x3f/0x60 [<ffffffff8140b84f>] notifier_call_chain+0x3f/0x80
[<ffffffff8140b8b5>] atomic_notifier_call_chain+0x15/0x20
[<ffffffff814044d4>] panic+0xf2/0x20f [<ffffffff810af362>]
watchdog_overflow_callback+0xd2/0xe0 [<ffffffff810c63b8>]
__perf_event_overflow+0xa8/0x220 [<ffffffff8101204f>] ?
x86_perf_event_set_period+0xdf/0x170 [<ffffffff810c6884>] perf_event_overflow
+0x14/0x20 [<ffffffff81017991>] intel_pmu_handle_irq+0x1a1/0x350
[<ffffffff81409639>] perf_event_nmi_handler+0x19/0x20 [<ffffffff81408e28>]
default_do_nmi+0x78/0x2d0 [<ffffffffa0a9e020>] ? yh_exit+0x20/0x20
[hard_lockup] [<ffffffff81409128>] do_nmi+0xa8/0xe0 [<ffffffff8140844c>]
end_repeat_nmi+0x1a/0x1e [<ffffffffa0a9e020>] ? yh_exit+0x20/0x20
[hard_lockup] [<ffffffff811f7daf>] ? delay_tsc+0x4f/0x90
[<ffffffff811f7daf>] ? delay_tsc+0x4f/0x90 [<ffffffff811f7daf>] ? delay_tsc
+0x4f/0x90 <<EOE>>  [<ffffffffa0a9e020>] ? yh_exit+0x20/0x20 [hard_lockup]
[<ffffffff811f7e4f>] __const_udelay+0x2f/0x40 [<ffffffffa0a9e07a>] yh_init
+0x5a/0xfe0 [hard_lockup] [<ffffffff810001cd>] do_one_initcall+0x3d/0x180
[<ffffffff81092c95>] sys_init_module+0xc5/0x220 [<ffffffff8140e4f9>]
system_call_fastpath+0x16/0x1b </kernel_stack> </pid:7344> <pid:
7344:7344:insmod> <basic> insmod         R  running 1      0  7344 7344 5706
NA-e NA-y NA-o (NOTLB) </basic> <vma> 00400000-00402000 r-xp 00000000 00:01
11732                              /sbin/insmod 00601000-00602000 r--p
00001000 00:01 11732                                  /sbin/insmod
00602000-00603000 rw-p 00002000 00:01 11732                                /
sbin/insmod 00603000-00624000 rw-p 00603000 00:00
0                                  [heap] 7f7258b44000-7f7258cb1000 r-xp
00000000 00:01 3173                       /lib64/libc-2.11.3.so
7f7258cb1000-7f7258eb1000 ---p 0016d000 00:01 3173                         /
lib64/libc-2.11.3.so 7f7258eb1000-7f7258eb5000 r--p 0016d000 00:01
3173                       /lib64/libc-2.11.3.so 7f7258eb5000-7f7258eb6000 rw-
p 00171000 00:01 3173                          /lib64/libc-2.11.3.so
7f7258eb6000-7f7258ebb000 rw-p 7f7258eb6000 00:00 0 7f7258ebb000-7f7258eda000
r-xp 00000000 00:01 3283                         /lib64/ld-2.11.3.so
7f72590d1000-7f72590d4000 rw-p 7f72590d1000 00:00 0 7f72590d8000-7f72590d9000
rw-p 7f72590d8000 00:00 0 7f72590d9000-7f72590da000 r--p 0001e000 00:01
3283                       /lib64/ld-2.11.3.so 7f72590da000-7f72590db000 rw-p
```



```
0001f000 00:01 3283                       /lib64/ld-2.11.3.so
7f72590db000-7f72590dc000 rw-p 7f72590db000 00:00 0 7fff72500000-7fff72522000
rw-p 7fffffdd000 00:00 0                          [stack]
7fff725b1000-7fff725b2000 r-xp 7fff725b1000 00:00 0 </vma> //Command line
information of the current process: <cmdline> insmod hard_lockup.ko </
cmdline> //Environment variables <env> LESSKEY=/etc/lesskey.bin
NNTPSERVER=news INFODIR=/usr/local/info:/usr/share/info:/usr/info MANPATH=
HOSTNAME=Storage XKEYSYMDB=/usr/X11R6/lib/X11/XKeysymDB HOST= TERM=xterm
SHELL=/bin/bash PROFILEREAD=true HISTSIZE=1000 SSH_CLIENT=128.5.64.161 54825
22 MORE=-sl SSH_TTY=/dev/pts/0 USER=admin
LS_COLORS=no=00:fi=00:di=01;34:ln=00;36:pi=40;33:so=01;35:do=01;35:bd=40;33;01
:cd=40;33;01:or=41;33;01:ex=00;32:*.cmd=00;32:*.exe=01;32:*.com=01;32:*.bat=01
;
32:*.btm=01;32:*.dll=01;32:*.tar=00;31:*.tbz=00;31:*.tgz=00;31:*.rpm=00;31:*.d
eb=00;31:*.arj=00;31:*.taz=00;31:*.lzh=00;31:*.lzma=00;31:*.zip=00;31:*.zoo=00
;
31:*.z=00;31:*.Z=00;31:*.gz=00;31:*.bz2=00;31:*.tb2=00;31:*.tz2=00;31:*.tbz2=0
0;31:*.avi=01;35:*.bmp=01;35:*.fli=01;35:*.gif=01;35:*.jpg=01;35:*.jpeg=01;35:
*.mng=01;35:*.mov=01;35:*.mpg=01;35:*.pcx=01;35:*.pbm=01;35:*.pgm=01;35:*.png=
01;35:*.ppm=01;35:*.tga=01;35:*.tif=01;35:*.xbm=01;35:*.xpm=01;35:*.dl=01;35:*
.gl=01;35:*.wmv=01;35:*.aiff=00;32:*.au=00;32:*.mid=00;32:*.mp3=00;32:*.ogg=00
;32:*.voc=00;32:*.wav=00;32: XNLSPATH=/usr/X11R6/lib/X11/nls ENV=/etc/
bash.bashrc HOSTTYPE=x86_64 FROM_HEADER= PAGER=less CSHEDIT=emacs
XDG_CONFIG_DIRS=/etc/xdg MINICOM=-c on MAIL=/var/mail/admin PATH=/sbin:/usr/
sbin:/usr/local/sbin:/OSM/bin:/OSM/bin/script:/OSM/script:/home/permitdir/
bin:/bin:/usr/bin:/usr/local/bin:/usr/games:/usr/lib/mit/bin:/usr/lib/mit/
sbin CPU=x86_64 INPUTRC=/etc/inputrc PWD=/startup_disk/conf/euler-kbox-220
LANG=POSIX TEXINPUTS=:/home/permitdir/.TeX:/usr/share/doc/.TeX:/usr/doc/.TeX
SHLVL=1 HOME=/home/permitdir LESS_ADVANCED_PREPROCESSOR=no OSTYPE=linux
LS_OPTIONS=-A -N --color=tty -T 0 WINDOWMANAGER= LESS=-M -I MACHTYPE=x86_64-
suse-linux-gnu LOGNAME=admin XDG_DATA_DIRS=/usr/share LC_CTYPE=en_US.UTF-8
SSH_CONNECTION=128.5.64.161 54825 128.5.110.220 22 LESSOPEN=lessopen.sh %s
INFOPATH=/usr/local/info:/usr/share/info:/usr/info LESSCLOSE=lessclose.sh %s
%s COLORTERM=1 OLDPWD=/startup_disk/conf _=/sbin/insmod </env> //Register
information: <pt_regs> RIP: 0033:[<00007f7258c1e35a>] RSP: 002b:
00007fff72520e58  EFLAGS: 00010202 RAX: 00000000000000af RBX:
ffffffff8140e4f9 RCX: 00007f7258c10130 RDX: 0000000000603010 RSI:
0000000000000e9f RDI: 0000000000603030 RBP: 0000000000004000 R08:
00007fff72520f70 R09: 00007f7258c58c60 R10: 00007f7258c10130 R11:
0000000000000202 R12: 0000000000603010 R13: 0000000000004000 R14:
0000000000000e9f R15: 0000000000603030 FS:  00007f72590d2700(0000)
GS:ffff880067820000(0000) knlGS:0000000000000000 CS:  0010 DS: 0000 ES: 0000
CR0: 0000000080050033 CR2: 00007f7258c1e350 CR3: 00000000368c8000 CR4:
00000000000407e0 </pt_regs>
```

2. Message output logs of the kernel

```
message: *****area type:message - location in message area:2***** //Time when
kernel panic occurs. The time is in UTC format. Beijing time = UTC time + 8
hours; panic time:20080120231731-a7634
[_main]: create_bond4 bond0 ... <6>[ 24.448240] bonding: bond0: releasing
active interface eth1 <1>[ 24.534488] [os_mgtibc_netconfig.rc] config mgt
ipv6 ip ... <1>[ 24.544442] [os_mgtibc_netconfig.rc] eth2 use default ipv6.
<5>[ 24.561750] [6073][5400040600ce][INF][Startup mode value=40][BSP]
[COM_Star.eQuery,362][cat] <6>[ 24.579933] Extended CMOS year: 2000
<6>[ 24.583514] Extended CMOS year: 2000 <5>[ 24.616672] [6087][5400040607da]
[INF][Current bios channel is : [1].][BSP][SP5_GetB.ersion,1386][cat]
<5>[ 24.635688] [6092][5400040607e3][INF][SPV2R1 BIOS Date:2013-07-12.][BSP]
[SPV2R1_G.osDate,1544][cat] <4>[ 24.654536] boot disk is /dev/sda
<1>[ 24.694332] [os_netmgt_main]: create_ifcfg6 eth2 ... <1>[ 24.754426]
[/etc/hotplug/os_mgtibc_netconfig.rc],OS_COPY_IP_FILE ifconfig-eth2/ip6g not
exist <1>[ 24.766967] [os_mgtibc_netconfig.rc] ------call
product_set_firewall.sh .------ <1>[ 24.792468] manage net is eth2.
<6>[ 24.800258] ip_tables: (C) 2000-2006 Netfilter Core Team <6>[ 24.815956]
nf_conntrack version 0.5.0 (16384 buckets, 65536 max) <6>[ 24.877504]
ip6_tables: (C) 2000-2006 Netfilter Core Team <6>[ 32.728790]
ADDRCONF(NETDEV_UP): eth2: link is not ready <4>[ 32.734510] ip used greatest
stack depth: 3840 bytes left <6>[ 34.332129] e1000e: eth2 NIC Link is Up 100
Mbps Full Duplex, Flow Control: None <6>[ 34.339509] e1000e 0000:34:00.0:
eth2: 10/100 speed: disabling TSO <6>[ 34.346105] ADDRCONF(NETDEV_CHANGE):
eth2: link becomes ready <6>[ 37.306582] bonding: bond0: Adding slave eth1.
```



```
<6>[ 37.389690] bonding: bond0: enslaving eth1 as an active interface with a
down link. <6>[ 38.890611] NET: Registered protocol family 17 <1>[ 38.930757]
----call maintain ip effect---- <1>[ 38.969397] [os_mgtibc_netconfig.rc] ----
Call maintain ip---- <1>[ 38.979805] [os_mgtibc_netconfig.rc] config maintain
ip ... <1>[ 39.009213] [os_netmgt_main]: get_ip4 eth2:1 ... <1>[ 39.071508]
[os_netmgt_main]: get_pf4 eth2:1 ... <1>[ 39.133710] [os_netmgt_main]:
get_gw4 eth2:1 ... <1>[ 39.196157] [os_netmgt_main]: get_bc4
eth2:1 ... ...... ......
```

3. Message output logs of the console

```
console: *****area type:console - location in console area:2***** //Time when
kernel panic occurs. The time is in UTC format. Beijing time = UTC time + 8
hours; panic time:20080120231731-a7634 //Information about the kernel stack:
[  392.145269]  ffff880061163c58 ffffffff811f52a9 0000000000000000
ffff8800e1163e27 [  392.152675]  ffff880061163d18 ffffffff811f5611
0000000000000292 ffff880061163c98 [  392.160075]  ffff880061163c0a
ffffffff0000ffff 0000000000000006 00ffffff819994a0 //Track of called kernel
functions: [  392.167477] Call Trace: [  392.169912]  [<ffffffff811f52a9>] ?
put_dec+0x59/0x60 [  392.174934]  [<ffffffff811f5611>] ? number+0x2f1/0x320
[  392.180044]  [<ffffffff811f5611>] ? number+0x2f1/0x320 [  392.185155]
[<ffffffff811f6a87>] ? vsnprintf+0x1d7/0x5b0 [  392.190524]
[<ffffffff81039166>] ? console_unlock+0x246/0x2a0 [  392.196325]
[<ffffffff8102e5a0>] ? __cpa_flush_all+0x60/0x60 [  392.202047]
[<ffffffffa0a9e020>] ? yh_exit+0x20/0x20 [hard_lockup] [  392.208278]
[<ffffffff814046b4>] ? printk+0x3c/0x40 [  392.213213]
[<ffffffff811f7db4>] ? delay_tsc+0x54/0x90 [  392.218410]
[<ffffffffa0a9e020>] ? yh_exit+0x20/0x20 [hard_lockup] [  392.224641]
[<ffffffff811f7e4f>] ? __const_udelay+0x2f/0x40 [  392.230270]
[<ffffffffa0a9e07a>] ? yh_init+0x5a/0xfe0 [hard_lockup] [  392.236588]
[<ffffffff810001cd>] ? do_one_initcall+0x3d/0x180 [  392.242389]
[<ffffffff81092c95>] ? sys_init_module+0xc5/0x220 [  392.248188]
[<ffffffff8140e4f9>] ? system_call_fastpath+0x16/0x1b [  392.255023] calling
kbox_sync :begin [  392.258576] sync kbox :begin [  392.261438] open all
redirect device :begin [  392.265596] open all redirect device :end
[  392.269580] flush kbox regions :begin [  392.273223] kbox region (panic)
is writing into (hmem), action is 202 [  392.280057] test write len : 6590
[  392.283352] first start addr : ffff88007e586000 [  392.287854] second
start addr : ffff88007e58e000 [  392.292444] cur_index : 2, offset : 32768,
third start addr : ffff88007e58e008 [  392.299623] first length : 6590
[  392.302766] bios_write_file_data write data len *ptr_length : 6590
[  392.308908] cur_index : 3, record_number : 3, total_number : 16
[  392.314791] kbox region (panic) has been written into (hmem)
[  392.320847] dev hmem is dirty //Time of occurrence panic time:
20080120231731-a7634
```

7.2.3 Providing System OOM Information

This topic describes exception information recorded by the kbox when the memory is insufficient.

Function Introduction

Due to special causes, the memory of the product or platform service software or OS may be insufficient, resulting in the OOM event. The kbox can record the OOM event information such as the time of occurrence, information about the process that encounters the OOM event, and system process information, in the storage device, and therefore facilitating fault locating.

Information List

The recorded OOM information includes three parts.

1. OOM exception information. Up to 256 KB information can be recorded. This area mainly records the following information:

- Time of occurrence (UTC time)
- PID and name of the current process, kernel version, and CPU number
- Track of the called abnormal process and stack information (up to 150 lines of the stack information can be output)
- Memory statistics
- Slab allocation information
- Information about current processes in the system
- Vmalloc information
- File information of the memory file systems such as rootfs, ramfs, and tmpfs
- Information about memory resources consumed by files in the memory file systems such as rootfs, ramfs, and tmpfs
- Actions after the OOM event (0: no operation; 1: calling the panic function)

2. Message output logs of the kernel. Up to 64 KB information can be recorded. This area mainly records the following information:
   - Time of occurrence (UTC time)
   - 64 KB of latest logs generated when the exception occurs, that is, the last 64 KB logs output by the kernel to the circular buffer
3. Message output logs of the console. Up to 32 KB information can be recorded. This area mainly records the following information:
   - Time of occurrence (UTC time)
   - Message output logs of other kernels

Examples

A log file includes the following information:

1. OOM exception information

```
//OOM exception storage area, the current OOM log is located in the OOM
exception storage area with index 1 *****area type:oom - location in oom area:
1***** ---------KBOX_START---------- //Time when kernel panic occurs. The
time is in UTC format. Beijing time = UTC time + 8 hours oom time:
20080121145818-505d5 //PID and name of the current process that triggers the
OOM process, kernel version, and CPU number Current Pid: 24527, comm:
exhaustmem Tainted: P           O 3.4.24.04-0.1-default #1 Stack:
ffff880130467ea0 ffffffffa0015177 ffffffffa00e5740 ffffffffa00d7cb0
00000000000005a0 ffffffffa00e5b98 ffff880130467ef0 ffffffffa00d3aa4
ffff880130467ec0 ffffffffa00d7c60 ffff880130467ef0 ffffffffa00e5740
ffffffffa00e5760 ffff880106768000 431bde82d7b634db ffff880106768000
ffff880130467f40 ffffffffa00d3e1c ffff880c4ee54700 0000000000000000
ffff8801064bd000 0000000000000000 0000000000000000 0000000000000000
0000000000000000 0000000000000000 0000000000000000 ffffffff8145a5d4 //Call
Trace: [<ffffffffa0000eea>] kbox_dump_oom+0x1a/0xb0 [kbox]
[<ffffffffa000b99c>] ? kbox_buffer_write+0xcc/0x110 [kbox]
[<ffffffffa000c71b>] ? kbox_message_duplicate_syslog+0x6b/0xb0 [kbox]
[<ffffffffa0000471>] kbox_oom_callback+0xe1/0x1e0 [kbox]
[<ffffffff814046b4>] ? printk+0x3c/0x40 [<ffffffff8140b84f>]
notifier_call_chain+0x3f/0x80 [<ffffffff8105b8c8>]
__blocking_notifier_call_chain+0x58/0x80 [<ffffffff8105b901>]
blocking_notifier_call_chain+0x11/0x20 [<ffffffff810ce65b>] out_of_memory
+0x4b/0x210 [<ffffffff810d3dd0>] __alloc_pages_slowpath+0x730/0x770
[<ffffffff810d4022>] __alloc_pages_nodemask+0x212/0x220 [<ffffffff81108e0a>]
alloc_pages_vma+0xba/0x190 [<ffffffff810e95eb>] do_anonymous_page+0x13b/0x2e0
[<ffffffff810ee098>] handle_pte_fault+0x1d8/0x1e0 [<ffffffff81407c99>] ?
_raw_spin_unlock+0x9/0x10 [<ffffffff810ee1e7>] handle_mm_fault+0x147/0x1d0
[<ffffffff8140b577>] do_page_fault+0x1f7/0x490 [<ffffffff810f2ff0>] ? do_brk
+0x260/0x350 [<ffffffff814080f5>] page_fault+0x25/0x30 //MemTotal:
```



```
1633712 kB MemFree:        1960 kB TotalHigh:          0 kB
FreeHigh:           0 kB Buffers:         108 kB SwapTotal:           0 kB
SwapFree:           0 kB //slabinfo: # name            <active_objs>
<num_objs> <objsize> <objperslab> <pagesperslab>: tunables <limit>
<batchcount> <sharedfactor>: slabdata <active_slabs> <num_slabs>
<sharedavail> nf_conntrack_expect      0      0    232 17    1 : tunables
120 60    8 : slabdata      0      0      0 nf_conntrack_ffffffff8184a680
73    120    328 12    1 : tunables 54 27    8 : slabdata     10     10
0 ext3_inode_cache      72     72    952    4    1 : tunables 54 27    8 :
slabdata     18     18      0 ext3_xattr           0      0     88 44    1 :
tunables  120 60    8 : slabdata      0      0      0 journal_handle
32    144     24  144    1 : tunables  120 60    8 : slabdata      1
1      0 journal_head         337 1360    112 34    1 : tunables  120 60
8 : slabdata     40     40    272 revoke_table           8    202     16
202    1 : tunables  120 60    8 : slabdata      1      1      0
revoke_record          0      0     32  112    1 : tunables  120 60    8 :
slabdata      0      0      0 uhci_urb_priv          0      0     56 67
1 : tunables  120 60    8 : slabdata      0      0      0
sd_ext_cdb             2    112     32  112    1 : tunables  120 60    8 :
slabdata      1      1      0 scsi_sense_cache      71     90  128 30    1 :
tunables  120 60    8 : slabdata      3      3      0 scsi_cmd_cache
45     45    256 15    1 : tunables  120 60    8 : slabdata      3
3      0 sgpool-128             3      3 4096    1    1 : tunables 24 12
8 : slabdata      3      3      0 sgpool-64              2      2 2048
2    1 : tunables 24 12    8 : slabdata      1      1      0
sgpool-32              2      4 1024    4    1 : tunables 54 27    8 :
slabdata      1      1      0 sgpool-16              2      8    512    8
1 : tunables 54 27    8 : slabdata      1      1      0 sgpool-8
30     30    256 15    1 : tunables  120 60    8 : slabdata      2
2      0 scsi_data_buffer       0      0     24  144    1 : tunables  120
60    8 : slabdata      0      0      0 RXRPC                  0      0
1152    7    2 : tunables 24 12    8 : slabdata      0      0      0
rxrpc_call_jar         0      0    960    4    1 : tunables 54 27    8 :
slabdata      0      0      0 fib6_nodes            11     59     64 59
1 : tunables  120 60    8 : slabdata      1      1      0
ip6_dst_cache          9     24    320 12    1 : tunables 54 27    8 :
slabdata      2      2      0 ...... ...... //Current process of the system:
[ pid ] uid  ppid  tgid total_vm      rss cpu stat name [    1]     0
0     1     2635      212 1    S                           init [  170]
0     1 170     6165      215 2    S                       os_exeshell
[ 1022]     0     1  1022     5688      337 3    S
vsftpd [ 1064]     0     1  1064     2086      217 0    S
os_sync_coffer [ 1081]     0     1  1081    16245      384 1
S                           sshd [ 3263]     0     1  3263     2280
154 0    S                      irqbalance [ 4424]     0     1  4424
2586      217 0    S                           syslogd [ 4427]     0     1
4427     2028     1151 2    D                            klogd [ 4439]
0     1  4439     4346      257 2    S                          xinetd
[ 4451]     0     1  4451     4703      219 1
S                           cron [ 4538] 105     1  4538     9022     1378
0    S                          dhcpd [ 6568]     0     1  6568
2925      485 1    S                os_sync_daemoni [ 6624]     0     1
6624     2925      488 3    S                   os_savetop.sh [ 6651]
0     1  6651     2925      480 1    S                    os_mgetty.sh
[ 6656]     0  6651  6656    15163      465 3    S
login [ 9314]     0  6656  9314     3979      536 0
S                           bash [11038]     0  1081 11038    23698
929 0    S                           sshd [11082]     0 11038 11082
4108      677 3    S                           bash [15647]     0  1081
15647    23699      935 0    S                           sshd [15656]     0
15647 15656     4108      679 1    S                           bash
[18411]     0  1081 18411    23699      936 1
S                           sshd [18561]     0 18411 18561     4108
678 2    S                           bash [22594]     0  6624 22594
1001      135 1    S                        sleep [24522]     0  6568
24522     1001      133 3    S                           sleep [24523]     0
15656 24523      998      123 3    S                        exhaustmem
[24524]     0 24523 24524    78338    77404 1    D
exhaustmem [24525]     0 24523 24525    78669    77722 0
```



```
D                        exhaustmem [24526]     0 24523 24526    78471
77546 2    D                        exhaustmem [24527]     0 24523 24527
78735    77812 3    R                        exhaustmem total real task
number: 29 *****************Start oom extend info.*****************
Vmallocinfo Start >>>>>>>>>>>>>>>>>>>>//Memory applied over the kernel
interface: 8192     hpet_enable+0x1d6/0x217 phys=0xfed00000 ioremap 4198400
alloc_large_system_hash+0x163/0x227 pages=1024 vmalloc vpages 12288
alloc_large_system_hash+0x163/0x227 pages=2 vmalloc 2101248
alloc_large_system_hash+0x163/0x227 pages=512 vmalloc 8192
acpi_os_map_memory+0xfb/0x15a phys=0x7ae5e000 ioremap 8192
acpi_os_map_memory+0xfb/0x15a phys=0x7aff8000 ioremap 8192
acpi_os_map_memory+0xfb/0x15a phys=0xfed1f000 ioremap 8192
acpi_os_map_memory+0xfb/0x15a phys=0x98521000 ioremap 53248
acpi_os_map_memory+0xfb/0x15a phys=0x7afeb000 ioremap 8192
acpi_os_map_memory+0xfb/0x15a phys=0x7afe9000 ioremap 8192
acpi_os_map_memory+0xfb/0x15a phys=0x7affb000 ioremap ...... Vmallocinfo End
<<<<<<<<<<<<<<<<<<<

 //Filesystem            1K-blocks    Used Available Use(%) Mounted on
rootfs                 1439256 189668  1249588       13% /
tmpfs                   756936    72072 684864        9% /dev/shm
tmpfs                   756936 144136 612800       19% /mnt ********    mem
info     ********       Total:                1513872 kB Total
free:            844252 kB User space:            523436 kB
Mlock:                      0 kB Kernel space:          146184 kB Bootmem
reserved:     2680372 kB HugePages_Total:           0 kB
HugePages_Free:             0 kB HugePages_Rsvd:              0 kB
HugePages_Surp:             0 kB Hugepagesize:            2048 kB ******
pagecache_info:     ******//Files occupying memory resources, from large to
small: /mnt/1 : nrpages = 18017. /dev/shm/3 : nrpages = 18017. /mnt/2 :
nrpages = 18017. //usr/share/cracklib/pw_dict.pwd : nrpages = 2534. //lib/
modules/3.4.24.19-0.11-default/kernel/vos/vos.ko : nrpages = 1656. //OSM/
modules/bsp.ko : nrpages = 1205. //lib/modules/3.4.24.19-0.11-default/kernel/
kbox/kbox.ko : nrpages = 1002. //usr/bin/ld.bfd : nrpages = 957. //usr/lib64/
libopcodes-2.23.1.so : nrpages = 774. //usr/lib64/libbfd-2.23.1.so : nrpages
= 774. //tmp/dump_27722 : nrpages = 587. //usr/sbin/dhcpd : nrpages =
515. //usr/lib64/libhd.so.15.50 : nrpages = 486. //lib/modules/3.4.24.19-0.11-
default/kernel/os_tools/osp_proc/osp_proc.ko : nrpages = 479. //bin/vim-
normal : nrpages = 453. //usr/lib64/libpython2.7.so.1.0 : nrpages =
453. //usr/Euler/conf/System.map : nrpages = 441. //OSM/conf/System.map :
nrpages = 441. //lib/libc-2.11.3.so : nrpages = 421. //usr/lib64/libxml2.so.
2.7.6 : nrpages = 404. //usr/share/misc/magic.mgc : nrpages = 401.

 ...... *****************End oom extend info.***************** //Actions
performed after the OOM exception occurs (0: return to the call chain; 1:
call the panic function) action after oom is:1 ---------
KBOX_END------------
```

2. Message output logs of the kernel

```
message: *****area type:message - location in message area:1***** oom time:
20080121145818-505d5 768(DMA)        0      0  32768    1    8 : tunables
8    4    0 : slabdata      0      0      0 <0>[ 2455.539133]
<4>[ 2455.539135] size-32768             2      2  32768    1    8 :
tunables    8    4    0 : slabdata      2      2      0 <0>[ 2455.539139]
<4>[ 2455.539141] size-16384(DMA)        0      0  16384    1    4 :
tunables    8    4    0 : slabdata      0      0      0 <0>[ 2455.539144]
<4>[ 2455.539148] size-16384            14     14  16384    1    4 :
tunables    8    4    0 : slabdata     14     14      0 <0>[ 2455.539152]
<4>[ 2455.539154] size-8192(DMA)         0      0 8192    1    2 :
tunables    8    4    0 : slabdata      0      0      0 <0>[ 2455.539157]
<4>[ 2455.539161] size-8192             17     17 8192    1    2 :
tunables    8    4    0 : slabdata     17     17      0 <0>[ 2455.539165]
<4>[ 2455.539167] size-4096(DMA)         0      0 4096    1    1 : tunables
24 12    8 : slabdata      0      0      0 <0>[ 2455.539171]
<4>[ 2455.539195] size-4096            349    352 4096    1    1 : tunables
24 12    8 : slabdata    349    352      0 <0>[ 2455.539198]
<4>[ 2455.539200] size-2048(DMA)         0      0 2048    2    1 : tunables
24 12    8 : slabdata      0      0      0 <0>[ 2455.539204]
<4>[ 2455.539243] size-2048            813    852 2048    2    1 : tunables
24 12    8 : slabdata    426    426      0 <0>[ 2455.539247]
```

```
<4>[ 2455.539249] size-1024(DMA)          0      0 1024    4    1 : tunables
54 27    8 : slabdata      0      0      0 <0>[ 2455.539252]
<4>[ 2455.539272] size-1024             902    960 1024    4    1 : tunables
54 27    8 : slabdata    240    240      0 <0>[ 2455.539276]
<4>[ 2455.539278] size-512(DMA)           0      0    512    8    1 : tunables
54 27    8 : slabdata      0      0      0 <0>[ 2455.539282]
<4>[ 2455.539288] size-512              367    392    512    8    1 : tunables
54 27    8 : slabdata     49     49     27 <0>[ 2455.539292]
<4>[ 2455.539294] size-256(DMA)           0      0    256 15    1 : tunables
120 60    8 : slabdata      0      0      0 <0>[ 2455.539298]
<4>[ 2455.539303] size-256              348    450    256 15    1 : tunables
120 60    8 : slabdata     30     30      0 <0>[ 2455.539307]
<4>[ 2455.539309] size-192(DMA)           0      0    192 20    1 : tunables
120 60    8 : slabdata      0      0  0 <0>[ 2455.539313] <4>[ 2455.539323]
size-192            1245 1320    192 20    1 : tunables  120 60    8 :
slabdata     66     66      0 <0>[ 2455.539326] <4>[ 2455.539328]
size-128(DMA)           0      0    128 30    1 : tunables  120 60  8 :
slabdata      0      0      0 <0>[ 2455.539332] <4>[ 2455.539334]
size-64(DMA)            0      0     64 59    1 : tunables  120 60    8 :
slabdata      0      0      0 <0>[ 2455.539338] <4>[ 2455.539340]
size-32(DMA)            0      0     32  112    1 : tunables  120 60    8 :
slabdata      0      0      0 <0>[ 2455.539344] <4>[ 2455.539370]
size-32             24432  24640     32  112    1 : tunables  120 60    8 :
slabdata    220    220      0 <0>[ 2455.539374] <4>[ 2455.539389]
size-128             3670 3840    128 30    1 : tunables  120 60    8 :
slabdata    128    128      0 <0>[ 2455.539393] <4>[ 2455.539449]
size-64             29752  29913     64 59    1 : tunables  120 60    8 :
slabdata    507    507     15 <0>[ 2455.539453] <4>[ 2455.539458]
kmem_cache            163    174    640    6    1 : tunables 54 27    8 :
slabdata     29     29      0 <0>[ 2455.539462] <0>[ 2455.540049]
<0>[ 2455.540051] mem info: <4>[ 2455.540077] MemTotal:        1633712 kB
<4>[ 2455.540078] MemFree:             1960 kB <4>[ 2455.540079]
Buffers:              108 kB <4>[ 2455.540080] Cached:           256004 kB
<4>[ 2455.540080] SwapCached:            0 kB <4>[ 2455.540081]
Active:          1262668 kB <4>[ 2455.540082] Inactive:            252 kB
<4>[ 2455.540083] Active(anon):    1262492 kB <4>[ 2455.540084]
Inactive(anon):       92 kB <4>[ 2455.540084] Active(file):        176 kB
<4>[ 2455.540085] Inactive(file):      160 kB <4>[ 2455.540086]
Unevictable:      255440 kB <4>[ 2455.540087] Mlocked:               0 kB
<4>[ 2455.540088] SwapTotal:             0 kB <4>[ 2455.540089]
SwapFree:                0 kB <4>[ 2455.540089] Dirty:                88 kB
<4>[ 2455.540090] Writeback:             0 kB <4>[ 2455.540091]
AnonPages:       1262516 kB <4>[ 2455.540092] Mapped:             4752 kB
<4>[ 2455.540093] Shmem:               108 kB <4>[ 2455.540093]
Slab:              44460 kB <4>[ 2455.540094] SReclaimable:      14908 kB
<4>[ 2455.540095] SUnreclaim:        29552 kB <4>[ 2455.540096]
KernelStack:       792 kB <4>[ 2455.540097] PageTables:         3904 kB
<4>[ 2455.540098] NFS_Unstable:          0 kB <4>[ 2455.540099]
Bounce:                0 kB <4>[ 2455.540099] WritebackTmp:          0 kB
<4>[ 2455.540100] CommitLimit:      816856 kB <4>[ 2455.540101]
Committed_AS:    1295624 kB <4>[ 2455.540102] VmallocTotal: 34359738367 kB
<4>[ 2455.540103] VmallocUsed:      275816 kB <4>[ 2455.540104] VmallocChunk:
34359461455 kB <4>[ 2455.540104] HardwareCorrupted:     0 kB
<4>[ 2455.540112] HugePages_Total:       0 <4>[ 2455.540113]
HugePages_Free:        0 <4>[ 2455.540114] HugePages_Rsvd:        0
<4>[ 2455.540114] HugePages_Surp:        0 <4>[ 2455.540115]
Hugepagesize:       2048 kB <4>[ 2455.540116] <4>[ 2455.540120]
DirectMap4k:      106496 kB <4>[ 2455.540121] DirectMap2M:     5136384 kB
<4>[ 2455.540121] DirectMap1G:    4296015872 kB <4>[ 2455.540122]
<0>[ 2455.540702] ps -aux: <4>[ 2455.540707] PID  PPID
COMMAND      VSZ      RSS             STAT <4>[ 2455.540712]     1
0                init 10540K     848K     S (sleeping) <4>[ 2455.540737]
2     0            kthreadd       0K       0K     S (sleeping)
<4>[ 2455.540740]     3     2          ksoftirqd/0       0K       0K     S
(sleeping) <4>[ 2455.540744]     4     2        kworker/0:0       0K
0K     S (sleeping) <4>[ 2455.540747]     6     2        os_watchdog/0
0K       0K     S (sleeping) <4>[ 2455.540750]     7     2          migration/
0       0K       0K     S (sleeping) <4>[ 2455.540763]     8     2
watchdog/0       0K       0K     S (sleeping) <4>[ 2455.540770]     9
```



```
2          migration/1       0K       0K     S (sleeping)
<4>[ 2455.540777]    10     2          kworker/1:0       0K       0K     S
(sleeping) <4>[ 2455.540783]    11     2        ksoftirqd/1       0K
0K     S (sleeping) <4>[ 2455.540790]    12     2        os_watchdog/1
0K       0K     S (sleeping) <4>[ 2455.540794]    13     2          kworker/
0:1       0K       0K     S (sleeping) <4>[ 2455.540809]    14
2           watchdog/1       0K       0K     S (sleeping)
<4>[ 2455.540823]    15     2          migration/2       0K       0K     S
(sleeping) <4>[ 2455.540830]    16     2          kworker/2:0       0K
0K     S (sleeping) <4>[ 2455.540857]    17     2          ksoftirqd/2
0K       0K     S (sleeping) <4>[ 2455.540875]    18     2        os_watchdog/
2       0K       0K     S (sleeping) <4>[ 2455.540897]    19     2
watchdog/2       0K       0K     S (sleeping) <4>[ 2455.540913]    20
2          migration/3       0K       0K     S (sleeping)
<4>[ 2455.540931]    21     2          kworker/3:0       0K       0K     S
(sleeping) <4>[ 2455.540948]    22     2          ksoftirqd/3       0K
0K     S (sleeping) <4>[ 2455.540974]    23     2        os_watchdog/3
0K       0K     S (sleeping) <4>[ 2455.540979]    24     2          watchdog/
3       0K       0K     S (sleeping) <4>[ 2455.541015]    25
2               cpuset       0K       0K     S (sleeping)
<4>[ 2455.541022]    26     2              khelper       0K       0K     S
(sleeping) <4>[ 2455.541061]    27     2            kdevtmpfs       0K
0K     S (sleeping) <4>[ 2455.541067]    28     2                netns
0K       0K     S (sleeping) <4>[ 2455.541090]  29     2
sync_supers       0K       0K     S (sleeping) <4>[ 2455.541159]    30
2          bdi-default       0K       0K     S (sleeping)
<4>[ 2455.541166]    31     2          kintegrityd       0K       0K     S
(sleeping) <4>[ 2455.541188]  32     2              kblockd       0K
0K     S (sleeping) <4>[ 2455.541220]    33     2          kworker/3:1
0K       0K     S (sleeping) <4>[ 2455.541233]    35     2          kworker/
2:1       0K       0K     S (sleeping) <4>[ 2455.541258]    36
2                 tifm       0K       0K     S (sleeping)
<4>[ 2455.541316]    37     2           khungtaskd       0K       0K     S
(sleeping) <4>[ 2455.541326]    38     2              kswapd0       0K
0K     S (sleeping) <4>[ 2455.541331]    39     2                ksmd
0K       0K     S (sleeping) <4>[ 2455.541372]    40     2
fsnotify_mark       0K       0K     S (sleeping) <4>[ 2455.541383]    41
2               crypto       0K       0K     S (sleeping)
<4>[ 2455.541415]    47     2             kthrotld       0K       0K     S
(sleeping) <4>[ 2455.541424]    61     2              deferwq       0K
0K     S (sleeping) <4>[ 2455.541435] 111     2              kauditd
0K       0K     S (sleeping) <4>[ 2455.541449] 157     2
vos       0K       0K     S (sleeping) <4>[ 2455.541455] 158     2
sched_work       0K       0K D (disk sleep) <4>[ 2455.541486] 159
2           sched_work       0K       0K D (disk sleep) <4>[ 2455.541506]
160     2           sched_work       0K       0K D (disk sleep)
<4>[ 2455.541513] 161     2           sched_work       0K       0K D (disk
sleep) <4>[ 2455.541517] 162     2           MML Server       0K       0K
S (sleeping) <4>[ 2455.541549] 163     2            diag_recv       0K
0K     S (sleeping) <4>[ 2455.541564] 170     1          os_exeshell
24660K     860K     S (sleeping) <4>[ 2455.541572] 280     2
ata_sff       0K       0K     S (sleeping) <4>[ 2455.541598] 282
2           scsi_eh_0       0K       0K     S (sleeping) <4>[ 2455.541605]
283     2           scsi_eh_1       0K       0K     S (sleeping)
<4>[ 2455.541620] 284     2            scsi_eh_2       0K       0K     S
(sleeping) <4>[ 2455.541706] 285     2            scsi_eh_3       0K
0K     S (sleeping) <4>[ 2455.541803] 286     2            scsi_eh_4
0K       0K     S (sleeping) <4>[ 2455.541836] 287     2
scsi_eh_5       0K       0K     S (sleeping) <4>[ 2455.541878] 289
2          kworker/u:3       0K       0K     S (sleeping) <4>[ 2455.541898]
290     2          kworker/u:4       0K       0K     S (sleeping)
<4>[ 2455.541956] 295     2                khubd       0K       0K     S
(sleeping) <4>[ 2455.541985] 335     2            kjournald       0K
0K     S (sleeping) <4>[ 2455.542026] 344     2          flush-259:0
0K       0K     S (sleeping) <4>[ 2455.542074] 359     2
kjournald       0K       0K     S (sleeping) <4>[ 2455.542158] 381
2            kjournald       0K       0K     S (sleeping) <4>[ 2455.542163]
428     2            kjournald       0K       0K     S (sleeping)
<4>[ 2455.542201] 645     2               bond0       0K       0K     S
```



```
(sleeping) <4>[ 2455.542211]  1022     1               vsftpd 22752K
1348K     S (sleeping) <4>[ 2455.542252]  1064     1       os_sync_coffer
8344K     868K     S (sleeping) <4>[ 2455.542284]  1081     1
sshd 64980K    1536K     S (sleeping) <4>[ 2455.542323]  1092     2
kworker/1:2       0K       0K     S (sleeping) <4>[ 2455.542345]  3263
1           irqbalance    9120K     616K     S (sleeping) <4>[ 2455.542405]
4424     1              syslogd 10344K     868K     S (sleeping)
<4>[ 2455.542416]  4427     1                klogd    8112K    4604K      R
(running) <4>[ 2455.542476]  4439     1               xinetd 17384K
1028K     S (sleeping) <4>[ 2455.542504]  4451     1                 cron
18812K     876K     S (sleeping) <4>[ 2455.542527]  4538     1
dhcpd 36088K    5512K     S (sleeping) <4>[ 2455.542577]  6568     1
os_sync_daemoni 11700K    1940K     S (sleeping) <4>[ 2455.542593]  6617
2         agetty_query       0K       0K D (disk sleep) <4>[ 2455.542602]
6624     1        os_savetop.sh 11700K    1952K     S (sleeping)
<4>[ 2455.542627]  6651     1         os_mgetty.sh 11700K    1920K     S
(sleeping) <4>[ 2455.542634]  6656  6651                login 60652K
1860K   S (sleeping) <4>[ 2455.542644]  9314  6656                 bash
15916K    2144K     S (sleeping) <4>[ 2455.542655] 11038
1081                 sshd 94792K    3716K     S (sleeping) <4>[ 2455.542662]
11082 11038                bash 16432K    2708K     S (sleeping)
<4>[ 2455.542676] 15647  1081                sshd 94796K    3740K     S
(sleeping) <4>[ 2455.542696] 15656 15647                 bash 16432K
2716K     S (sleeping) <4>[ 2455.542706] 18411  1081                 sshd
94796K    3744K     S (sleeping) <4>[ 2455.542714] 18561
18411                bash 16432K    2712K     S (sleeping) <4>[ 2455.542728]
22594  6624               sleep    4004K     540K     S (sleeping)
<4>[ 2455.542767] 24522  6568               sleep    4004K     532K     S
(sleeping) <4>[ 2455.542779] 24523 15656          exhaustmem    3992K
492K     S (sleeping) <4>[ 2455.542802] 24524 24523          exhaustmem
313352K  309616K D (disk sleep) <4>[ 2455.542838] 24525 24523
exhaustmem  314676K  310888K D (disk sleep) <4>[ 2455.542856] 24526
24523          exhaustmem  313884K  310184K D (disk sleep) <4>[ 2455.542868]
24527 24523          exhaustmem  314940K  311248K      R (running)
<4>[ 2455.543461] Mem-Info: <4>[ 2455.543482] Node 0 DMA per-cpu:
<4>[ 2455.543562] CPU    0: hi:    0, btch: 1 usd: 0 <4>[ 2455.543564] CPU
1: hi:    0, btch: 1 usd: 0 <4>[ 2455.543567] CPU    2: hi:    0, btch: 1
usd: 0 <4>[ 2455.543569] CPU    3: hi:    0, btch: 1 usd: 0 <4>[ 2455.543571]
Node 0 DMA32 per-cpu: <4>[ 2455.543573] CPU    0: hi:  186, btch:  31 usd: 0
<4>[ 2455.543576] CPU    1: hi:  186, btch:  31 usd: 0 <4>[ 2455.543578]
CPU    2: hi:  186, btch:  31 usd: 0 <4>[ 2455.543580] CPU    3: hi:  186,
btch:  31 usd: 0 <4>[ 2455.543585] active_anon:315623 inactive_anon:23
isolated_anon:0 <4>[ 2455.543586]  active_file:44 inactive_file:40
isolated_file:0 <4>[ 2455.543587]  unevictable:63860 dirty:22 writeback:0
unstable:0 <4>[ 2455.543588]  free:490 slab_reclaimable:3727
slab_unreclaimable:7388 <4>[ 2455.543589]  mapped:1188 shmem:27 pagetables:
976 bounce:0 <4>[ 2455.543592] Node 0 DMA free:24kB min:16kB low:20kB high:
24kB active_anon:15852kB inactive_anon:0kB active_file:0kB inactive_file:0kB
unevictable:0kB isolated(anon):0kB isolated(file):0kB present:15676kB mlocked:
0kB dirty:0kB writeback:0kB mapped:0kB shmem:0kB slab_reclaimable:0kB
slab_unreclaimable:0kB kernel_stack:0kB pagetables:0kB unstable:0kB bounce:
0kB writeback_tmp:0kB pages_scanned:0 all_unreclaimable? yes
<4>[ 2455.543603] lowmem_reserve[]: 0 2 3072 128 <4>[ 2455.543608] Node 0
DMA32 free:1936kB min:2028kB low:2532kB high:3040kB active_anon:1246640kB
inactive_anon:92kB active_file:176kB inactive_file:160kB unevictable:255440kB
isolated(anon):0kB isolated(file):0kB present:1764800kB mlocked:0kB dirty:
88kB writeback:0kB mapped:4752kB shmem:108kB slab_reclaimable:14908kB
slab_unreclaimable:29552kB kernel_stack:792kB pagetables:3904kB unstable:0kB
bounce:0kB writeback_tmp:0kB pages_scanned:1022 all_unreclaimable? yes
<4>[ 2455.543620] lowmem_reserve[]: 0 0 256 256 <4>[ 2455.543624] Node 0 DMA:
1*4kB 0*8kB 0*16kB 0*32kB 0*64kB 0*128kB 0*256kB 0*512kB 0*1024kB 0*2048kB
0*4096kB = 4kB <4>[ 2455.543634] Node 0 DMA32: 226*4kB 67*8kB 8*16kB 3*32kB
4*64kB 0*128kB 1*256kB 0*512kB 0*1024kB 0*2048kB 0*4096kB = 2176kB
<4>[ 2455.543645] 64028 total pagecache pages <4>[ 2455.543646] 0 pages in
swap cache <4>[ 2455.543649] Swap cache stats: add 0, delete 0, find 0/0
<4>[ 2455.543650] Free swap  = 0kB <4>[ 2455.543652] Total swap = 0kB
<4>[ 2455.548040] 487408 pages RAM <4>[ 2455.548043] 78981 pages reserved
<4>[ 2455.548044] 7204 pages shared <4>[ 2455.548046] 397079 pages non-shared
```

3. Logs generated by the console


```
console: *****area type:console - location in console area:1***** oom time:
20080121145818-505d5 [ 2455.550908] Mem-Info: [ 2455.553173] Node 0 DMA per-
cpu: [ 2455.556302] CPU    0: hi:    0, btch: 1 usd: 0 [ 2455.561064] CPU
1: hi:    0, btch: 1 usd: 0 [ 2455.565825] CPU    2: hi:    0, btch: 1 usd: 0
[ 2455.570586] CPU    3: hi:    0, btch: 1 usd: 0 [ 2455.575347] Node 0 DMA32
per-cpu: [ 2455.578651] CPU    0: hi:  186, btch:  31 usd: 0 [ 2455.583412]
CPU    1: hi:  186, btch:  31 usd: 0 [ 2455.588173] CPU    2: hi:  186,
btch:  31 usd: 0 [ 2455.592934] CPU    3: hi:  186, btch:  31 usd: 0
[ 2455.597698] active_anon:315623 inactive_anon:23 isolated_anon:0
[ 2455.597699]  active_file:44 inactive_file:40 isolated_file:0
[ 2455.597700]  unevictable:63860 dirty:22 writeback:0 unstable:0
[ 2455.597701]  free:490 slab_reclaimable:3727 slab_unreclaimable:7388
[ 2455.597703]  mapped:1188 shmem:27 pagetables:976 bounce:0 [ 2455.626584]
Node 0 DMA free:24kB min:16kB low:20kB high:24kB active_anon:15852kB
inactive_anon:0kB active_file:0kB inactive_file:0kB unevictable:0kB
isolated(anon):0kB isolated(file):0kB present:15676kB mlocked:0kB dirty:0kB
writeback:0kB mapped:0kB shmem:0kB slab_reclaimable:0kB slab_unreclaimable:
0kB kernel_stack:0kB pagetables:0kB unstable:0kB bounce:0kB writeback_tmp:0kB
pages_scanned:0 all_unreclaimable? yes [ 2455.662953] lowmem_reserve[]: 0 2
3072 128 [ 2455.667074] Node 0 DMA32 free:1936kB min:2028kB low:2532kB high:
3040kB active_anon:1246640kB inactive_anon:92kB active_file:176kB
inactive_file:0kB unevictable:255440kB isolated(anon):0kB isolated(file):0kB
present:1764800kB mlocked:0kB dirty:88kB writeback:0kB mapped:4752kB shmem:
108kB slab_reclaimable:14908kB slab_unreclaimable:29552kB kernel_stack:792kB
pagetables:3904kB unstable:0kB bounce:0kB writeback_tmp:0kB pages_scanned:
4640 all_unreclaimable? yes [ 2455.707242] lowmem_reserve[]: 0 0 256 256
[ 2455.711283] Node 0 DMA: 1*4kB 0*8kB 0*16kB 0*32kB 0*64kB 0*128kB 0*256kB
0*512kB 0*1024kB 0*2048kB 0*4096kB = 4kB [ 2455.721622] Node 0 DMA32: 226*4kB
67*8kB 8*16kB 3*32kB 4*64kB 0*128kB 1*256kB 0*512kB 0*1024kB 0*2048kB
0*4096kB = 2176kB [ 2455.732650] 64028 total pagecache pages [ 2455.736460] 0
pages in swap cache [ 2455.739752] Swap cache stats: add 0, delete 0, find
0/0 [ 2455.744944] Free swap  = 0kB [ 2455.747804] Total swap = 0kB
[ 2455.755068] 487408 pages RAM [ 2455.757938] 78981 pages reserved
[ 2455.761143] 7202 pages shared [ 2455.764089] 397079 pages non-shared
[ 2455.767672] calling kbox_sync :begin [ 2455.771223] sync kbox :begin
[ 2455.774085] open all redirect device :begin [ 2455.778242] open all
redirect device :end [ 2455.782224] flush kbox regions :begin [ 2455.785864]
kbox region (oom) is writing into (hmem), action is 202 [ 2455.792525] test
write len : 21748 [ 2455.795902] first start addr : ffff88007e686000
[ 2455.800403] second start addr : ffff88007e6a6000 [ 2455.804990]
cur_index : 1, offset : 131072, third start addr : ffff88007e6a6008
[ 2455.812252] first length : 21748 [ 2455.815508] bios_write_file_data write
data len *ptr_length : 21748 [ 2455.821735] cur_index : 2, record_number : 2,
total_number : 4 [ 2455.827538] kbox region (oom) has been written into
(hmem) [ 2455.833420] dev hmem is dirty oom time:20080121145818-505d5
```

7.2.4 Providing System Die or OOPS Information

This topic describes exception information recorded by the kbox when the memory encounters the OOPS exception.

Function Introduction

Due to special causes, the memory of the product or platform service software or OS may access a null pointer or be destroyed, resulting in the OOPS event. The kbox can record the OOPS event information such as the time of occurrence, information about the process that encounters the OOPS event, and call stack of the abnormal process, in the storage device, and therefore facilitating fault locating.

Information List

The recorded die or OOPS information includes three parts.

1. OOPS information. Up to 128 KB information can be recorded. This area mainly records the following information:

- Time of occurrence (UTC time)
- PID and name of the current process
- Track of the called abnormal process and stack information (up to 150 lines of the stack information can be output)
- Start address and address length of the module in the output information (up to 384 module records can be output)

NOTE

If a large number of modules exist in the system and the name of each module is long, log shock may occur.

2. Message output logs of the kernel. Up to 64 KB information can be recorded. This area mainly records the following information:
   - Time of occurrence (UTC time)
   - 64 KB of latest logs generated when the exception occurs, that is, the last 64 KB logs output by the kernel to the circular buffer
3. Message output logs of the console. Up to 32 KB information can be recorded. This area mainly records the following information:
   - Time of occurrence (UTC time)
   - Message output logs of other kernels

Examples

A log file includes the following information:

1. OOPS exception information

```
*****area type:die - location in die area:0***** //Error type and error
number: die info:Oops:0002 ---------KBOX_START---------- //die time:
20131109041507-e9ea9 //Number of the CPU where the process runs, and process
name: CPU 0 Pid: 7664, comm: bash //Register information RIP: 0010:
[<ffffffffa0adbcdf>]  [<ffffffffa0adbcdf>] dev_wr_handler+0xa3f/0xce0 [kpgen]
RSP: 0018:ffff88005ea73eb8  EFLAGS: 00010292 RAX: 0000000000000045 RBX:
0000000000000000 RCX: 0000000000006161 RDX: 0000000000000062 RSI:
0000000000000096 RDI: 0000000000000246 RBP: ffff88005ea73f08 R08:
ffff88007b001000 R09: 0000000000000000 R10: 0000000000000000 R11:
0000000000000000 R12: ffffffff810f10f0 R13: 0000000000000005 R14:
00007f8bcf81a000 R15: 0000000000000000 FS:  00007f8bcf97f700(0000)
GS:ffff880069600000(0000) knlGS:0000000000000000 CS:  0010 DS: 0000 ES: 0000
CR0: 0000000080050033 CR2: 0000000000000000 CR3: 0000000035c43000 CR4:
00000000001407f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2:
0000000000000000 DR3: 0000000000000000 DR6: 00000000ffff0ff0 DR7:
0000000000000400 Process bash (pid: 7664, threadinfo ffff88005ea72000, task
ffff8800373ba6c0) Stack: 0000000a32303031 0000000000000000 0000000000000000
0000000000000000 ffff88005ea73f08 ffffffff81146a08 0000000000000002
0000000000000005 ffff88004f3b8e80 ffff88005ea73f48 ffff88005ea73f38
ffffffff81146e9b //Call Trace: [<ffffffffa0003e43>] kbox_show_registers
+0x4c3/0xa30 [kbox] [<ffffffffa000e91b>] ? kbox_buffer_write+0xcb/0x110
[kbox] [<ffffffffa001012c>] kbox_die_callback+0x15c/0x220 [kbox]
[<ffffffff81445e0f>] notifier_call_chain+0x3f/0x80 [<ffffffff81445e5d>]
__atomic_notifier_call_chain+0xd/0x10 [<ffffffff81445e71>]
atomic_notifier_call_chain+0x11/0x20 [<ffffffff81445eae>] notify_die+0x2e/
0x30 [<ffffffff814430c8>] __die+0x88/0x100 [<ffffffff8102e883>] no_context
+0xf3/0x200 [<ffffffff8102eabd>] __bad_area_nosemaphore+0x12d/0x220
[<ffffffff8102ec09>] bad_area+0x49/0x60 [<ffffffff81445d6c>] do_page_fault
+0x44c/0x4b0 [<ffffffff8103aef6>] ? console_unlock+0x246/0x2a0
[<ffffffff810f10f0>] ? oom_kill_process+0x2a0/0x2a0 [<ffffffff81442675>]
page_fault+0x25/0x30 [<ffffffff810f10f0>] ? oom_kill_process+0x2a0/0x2a0
[<ffffffffa0adbcdf>] ? dev_wr_handler+0xa3f/0xce0 [kpgen]
[<ffffffffa0adbcdf>] ? dev_wr_handler+0xa3f/0xce0 [kpgen]
[<ffffffff81146a08>] ? rw_verify_area+0x58/0x100 [<ffffffff81146e9b>]
```



```
vfs_write+0xcb/0x130 [<ffffffff81146ff0>] sys_write+0x50/0x90
[<ffffffff8144a079>] system_call_fastpath+0x16/0x1b Code: 00 00 0f 85 12 f7
ff ff 48 c7 c7 08 d5 ad a0 31 c0 e8 58 2e 96 e0 cd 04 e9 0b f7 ff ff 48 c7 c7
98 d4 ad a0 31 c0 e8 43 2e 96 e0 <c6> 04 25 00 00 00 00 00 e9 f0 f6 ff ff 48
c7 c7 d0 d4 ad a0 31 mod_name                mod_start             core_size
os_test                 0xffffffffa0015000    0x3df6 bioshmem_driver
0xffffffffa000c000    0x454d kbox                    0xffffffffa0b81000
0x49a53 agetty_query            0xffffffffa0b7c000    0x3442
cpufreq_powersave     0xffffffffa0b77000    0x314b signo_catch
0xffffffffa0b6d000    0x30fa sysalarm_agent_netlink_k0xffffffffa0b72000
0x316d af_packet               0xffffffffa0b63000    0x8a5e ...... //Actions
performed after the die exception occurs (0: no action; 1: calling the panic
function; 2: reboot) action after die is:0 ---------KBOX_END------------
```

2. Message output logs of the kernel

```
message: *****area type:message - location in message area:4***** die time:
20131109041507-e9ea9
>[  246.509001] INFO: task sched_work:253 blocked for more than 120 seconds.
<3>[  246.509005] "echo 0 > /proc/sys/kernel/hung_task_timeout_secs" disables
this message. <6>[  246.509009] sched_work      D 0000000000000000  5544
253      2 0x00000000 <4>[  246.509018]  ffff88005eb0bdd0 0000000000000046
ffffffff81075026 ffff88005eb0a010 <4>[  246.509025]  00000000000159c0
00000000000159c0 00000000000159c0 00000000000159c0 <4>[  246.509032]
ffff88005eb0bfd8 ffff88005eb0bfd8 00000000000159c0 00000000000159c0
<4>[  246.509038] Call Trace: <4>[  246.509048]  [<ffffffff81075026>] ?
update_curr+0x186/0x1c0 <4>[  246.509055]  [<ffffffff8107607a>] ?
dequeue_task_fair+0x6a/0x170 <4>[  246.509060]  [<ffffffff8106ba29>] ?
dequeue_task+0x89/0xa0 <4>[  246.509082]  [<ffffffffa00bf7fe>] ? DBG_Log+0x3e/
0x340 [vos] <4>[  246.509090]  [<ffffffff81440499>] schedule+0x29/0x90
<4>[  246.509095]  [<ffffffff8143edad>] schedule_timeout+0x21d/0x2c0
<4>[  246.509102]  [<ffffffff810c1392>] ? call_rcu_sched+0x12/0x20
<4>[  246.509107]  [<ffffffff8106261a>] ? __put_cred+0x3a/0x50
<4>[  246.509111]  [<ffffffff81062d2e>] ? commit_creds+0x12e/0x1e0
<4>[  246.509117]  [<ffffffff8143f54d>] __down+0x6d/0xb0 <4>[  246.509125]
[<ffffffff81061aa7>] down+0x47/0x50 <4>[  246.509142]  [<ffffffffa00c6009>]
LVOS_sema_down+0x9/0x10 [vos] <4>[  246.509157]  [<ffffffffa00c64cc>]
LVOS_SchedWorkThread+0x5c/0x190 [vos] <4>[  246.509165]  [<ffffffff8144b3d4>]
kernel_thread_helper+0x4/0x10 <4>[  246.509180]  [<ffffffffa00c6470>] ?
LVOS_SchedWorkInit+0x60/0x60 [vos] <4>[  246.509186]  [<ffffffff8144b3d0>] ?
gs_change+0x13/0x13 <3>[  246.509190] INFO: task sched_work:254 blocked for
more than 120 seconds. <3>[  246.509192] "echo 0 > /proc/sys/kernel/
hung_task_timeout_secs" disables this message. <6>[  246.509196]
sched_work      D 0000000000000000  5544 254      2 0x00000000
<4>[  246.509203]  ffff88005eb07dd0 0000000000000046 ffffffff81075026
ffff88005eb06010 <4>[  246.509209]  00000000000159c0 00000000000159c0
00000000000159c0 00000000000159c0 <4>[  246.509214]  ffff88005eb07fd8
ffff88005eb07fd8 00000000000159c0 00000000000159c0 <4>[  246.509220] Call
Trace: <4>[  246.509225]  [<ffffffff81075026>] ? update_curr+0x186/0x1c0
<4>[  246.509231]  [<ffffffff8107607a>] ? dequeue_task_fair+0x6a/0x170
<4>[  246.509236]  [<ffffffff8106ba29>] ? dequeue_task+0x89/0xa0
<4>[  246.509249]  [<ffffffffa00bf7fe>] ? DBG_Log+0x3e/0x340 [vos]
<4>[  246.509256]  [<ffffffff81440499>] schedule+0x29/0x90 <4>[  246.509260]
[<ffffffff8143edad>] schedule_timeout+0x21d/0x2c0 <4>[  246.509265]
[<ffffffff810c1392>] ? call_rcu_sched+0x12/0x20 <4>[  246.509269]
[<ffffffff8106261a>] ? __put_cred+0x3a/0x50 <4>[  246.509274]
[<ffffffff81062d2e>] ? commit_creds+0x12e/0x1e0 <4>[  246.509279]
[<ffffffff8143f54d>] __down+0x6d/0xb0 <4>[  246.509286]  [<ffffffff81061aa7>]
down+0x47/0x50
```

3. Message output logs of the console

```
console: *****area type:console - location in console area:4***** die time:
20131109041507-e9ea9 die time:20131109041507-e9ea9
```

7.3 Kbox Operation Methods

This topic describes the operation process and precautions of the kbox.

7.3.1 Workflow

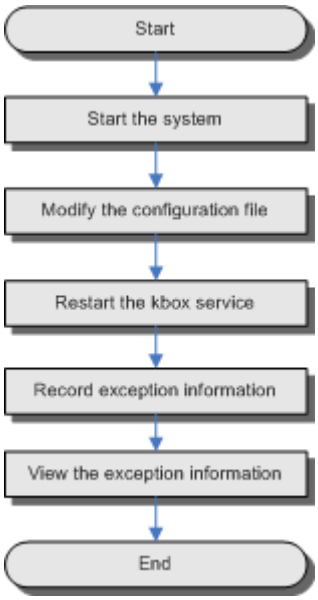
7.3.1 Workflow

This optic describes the workflows of the kbox, including modifying configuration files, restarting kbox services, and viewing exception information.

Workflow

Figure 7-2 describes the workflow of the kbox.

Figure 7-2 Simple workflow of the kbox



NOTE

1. In the EulerOS system, the kbox is delivered with the kernel together. The kbox service is automatically enabled when the system starts up.

2. You must configure the **reserve_mem** parameter in the startup parameters. By default, this parameter is set to **16M**. If this parameter is not configured or the value is smaller than 16 MB, the kbox service fails to be enabled.

Process

Table 7-2 describes the specific process.

NOTICE

- The storage medium must be configured. Otherwise, the captured information cannot be saved and will be lost after the system restart.

- After the configuration, you must restart the kbox service to validate the settings.

Table 7-2 Kbox workflow

SN	Process	Description
1	Start the system.	In the EulerOS system, the kbox is delivered with the kernel together. The kbox service is automatically enabled when the system starts up.
2	Modify the configuration file.	The product information, exception events to be captured, and available storage means are configured. The kbox configuration file is saved in the path **/etc/kbox/config**.
3	Restart the kbox service.	After the configuration file is modified, the kbox service shall be restarted to validate the settings.
4	Record exception information.	When an exception event occurs, the kbox records the specified exception information in the corresponding storage device based on the configuration file.
5	View the exception information.	The user with the root rights can run the log export command to export the exception information to a file in the specified directory from the storage device.

7.3.2 Modifying the Configuration File

This topic describes how to properly configure the kbox.

Prerequisites

- The kbox is running properly in the EulerOS.
- The dump medium is working properly.

Precautions

After modifying the configuration file, restart the kbox to validate the settings.

Configuration Procedure

In the shell command line environment, log in as a user with the root rights.

1. Open the configuration file. In any path, run the following command:
   ```
   vi  /etc/kbox/config
   ```
   The content in the configuration file is as follows:
   ```
   #This file is the configuration file of the kbox. #Configure the file based
   on the specific product information. #*Caution*: Correctly configure the
   parameters based on the prompt information. The parameter must be quoted
   using English quotation marks and no space exists inside or outside the
   quotation marks.

    #############  Basic information of the hardware platform  ##############
   #Product name, for example, "euler" "pangea" "dopra" "cgp" product="euler"

    #Product version No., for example, V100R001C00B251 version="V002R200C10"

    #Frame No., for example, frame=1~9 (generally for cascading) frame=""

    #Slot No. of the board, for example, slot=0~12 slot=""

    #Board location, for example, front board or rear board (0~1) locate=""

    #Hardware information of the board, for example, pangea-v2
   hard_version="pangea-v2"

    ############ Scenarios for capturing exception information. If the value is
   null or illegal, the system considers that the value is set to no
   ############## #panic event: panic triggered by the action of explicitly
   calling the panic function of the kernel or an exception, panic_event="yes"

    #Normal reboot event: The reboot, shutdown, halt, and init reboot_event="no"
   triggered by commands in user mode

    #Abnormal reboot event: The machine restart emerge_event="no" triggered by
   exceptions in kernel mode

    #Die event (optional), die_event="yes"

    #Actions performed after the die event, die_call_panic="no"

    #OOM event, oom_event="yes" #Whether to call panic oom_call_panic="no" after
   the OOM event occurs

    #D deadlock event, dlock_event="no"

    #R deadlock event, rlock_event="no"

    #Acpi event, to-be-defined, acpi_event="no"

    #Mce event, to-be-defined, mainly the intel CPU-related exception,
   mce_event="no"

    #BMC pre-interrupt event, mainly for the MCCP products, bmc_event="no"
   ```



```
 #User-defined event, adding or deleting ip ipaddr_event="no"

 #Network events, mainly the up/down event of the network interface,
net_event="no"

 #User-defined events: adding or deleting a route, adding or deleting a rule,
modification time (whether to separate the modification time is to be
determined), route_event="no" rule_event="no" time_event="no"

 #Security events: mainly the security hardening-related events,
security_event="no"

 #Watchdog events: mainly the watchdog expiration reset event,
wtdog_event="no"


 ##############  Exception storage medium  ################ #bmc device
bmc="off"

 #BIOS device, bios="off"

 #PCIE device, pcie="off"

 #Highmem device, hmem="on"

 #Logic device, logic="off"

 ################  Module insertion parameters  ############### #Note: The
start_pddr must be a physical address, aligned by 4 KB, in hexadecimal format
#      mem_size is in hexadecimal format, a minimum of 8 KB #      Parameters
are separated by spaces #bmc drive parameter bmc_param=""

 #BIOS drive parameter, bios_param=""

 #PCIE drive parameter, pcie_param=""

 #Highmem drive, hmem_param="sym_start_paddr_name=reserve_kbox_mem_start
sym_mem_size_name=reserve_kbox_mem_len"

 #Logic drive parameter, logic_param=""

 #############  Internal test devices  ############# #net device (for
internal use) net="off"

 ################  IP address of the TFTP server (for internal use)
################ #When the net device is set to the on state, the messages in
the region can be sent to the TFTP server. By default, messages of the TFTP
server with IP address 127.0.0.1 are not sent, TGT_IP="127.0.0.1" TGT_MAC=""

 ############  Crash report interface  ############# die_notify_func=""
```

2. Modify the configuration file.

   Configure basic information, exception events to be captured, and storage devices based on the current hardware environment of the system. The configuration mainly includes the product information, exception event, and storage device.

NOTICE

- The parameters must be configured in strict accordance with the requirements. The parameters must be quoted in English quotation marks without any spaces inside or outside the quotation marks. The space between characters must be replaced by "_". The capitalization of configuration items set to yes/no, YES/NO, on/off, or ON/OFF must be consistent. No space is allowed. Otherwise, the value is considered as **no**.

- At least one configuration item is configured for the exception event. Otherwise, the kbox cannot load the configuration file. The storage device must be configured. Otherwise, the exception information cannot be dumped. After the kbox is restarted, the information is lost.
- After modifying the configuration file, restart the kbox to validate the settings.

  Table 7-3, **Table 7-4**, and **Table 7-5** describe the parameters and configuration requirements.

Table 7-3 Product information

SN	Parameter Name	Description	Parameter Value	Mandatory or Not
1	product	Current device name (or product name)	User-defined character string	No
2	version	Version number of the current system	User-defined character string	No
3	frame	Current rack number	An integer larger than 0	No
4	slot	Slot number of the current board	An integer larger than 0	No
5	locate	Board location	0 or 1	No
6	hard_version	Hardware version	User-defined character string	No

Table 7-4 Exception events

SN	Parameter Name	Description	Parameter Value	Support or Not	Mandatory or Not
1	panic_event	Panic event	yes/no	Yes	Yes. At least one type of exception events shall be configured.
2	reboot_event	Security reboot event	yes/no	No	
3	emerge_event	Abnormal reboot event	yes/no	No	
4	die_event	Die event	yes/no	Yes	
5	dlock_event	D deadlock event	yes/no	No	

SN	Parameter Name	Description	Parameter Value	Support or Not	Mandatory or Not
6	rlock_event	R deadlock event	yes/no	No	
7	acpi_event	Power-on/off event	yes/no	No	
8	mce_event	Hardware error event	yes/no	No	
9	bmc_event	BMC pre-interrupt event	yes/no	No	
10	ipaddr_event	IP address modification event	yes/no	No	
11	net_event	Status change event of the Net device interface	yes/no	No	
12	route_event	Route modification event	yes/no	No	
13	rule_event	Change event of user-defined ACL rules	yes/no	No	
14	time_event	Time zone or time change event	yes/no	No	
15	security_event	User-defined security event	yes/no	No	
16	wtdog_event	Watchdog event	yes/no	No	
17	oom_event	OOM event	yes/no	Yes	
18	oom_call_panic	Whether to call the panic function after the OOM event occurs	yes/no	Yes	No

NOTE

The **oom_call_panic** configuration item takes effect only when the **oom_event** configuration item is set to **yes**.

Table 7-5 Storage device configurations

SN	Parameter Name	Description	Parameter Value	Mandatory or Not
1	bmc	BMC storage	on/off	Yes. At least one storage mode must be configured.
2	bios	BIOS NVRAM storage	on/off	
3	pcie	PCIE storage	on/off	
4	hmem	High-end memory storage	on/off	
5	logic	Logic storage	on/off	

NOTE

1. Table 3 and Table 4 must be used together.

2. The CloudEdge uses the high-end memory storage. By default, **hmem** is set to **on**.

Table 7-6 Storage configuration parameters

SN	Parameter Name	Description	Parameter Value	Mandatory or Not
1	bmc_param	BMC storage drive parameter	Not supported temporarily	No
2	bios_param	BIOS NVRAM drive parameter	Default parameter of the drive	No
3	pcie_param	PCIE drive parameter	pci_bus_id=00 pci_device_id=00	Yes

SN	Parameter Name	Description	Parameter Value	Mandatory or Not
4	hmem_param	Reserved memory drive parameter	This parameter supports two value configuration modes: ● sym_start_paddr_name=reserve_kbox_mem_start sym_mem_size_name=reserve_kbox_mem_len ● start_paddr=0xxxxxx mem_size=0xxxxx. The values of **start_paddr** and **mem_size** are integers in hexadecimal format. The **start_paddr** specifies the start physical address, and **mem_size** specifies the size of the memory. By default, the first value configuration mode is adopted. The CloudEdge user is not suggested to modify the value.	Yes
5	logic_param	Logic drive parameter	Not supported temporarily	No

NOTE

- The CloudEdge saves the memory information using the boot parameters instead of the non-volatile storage device. The common memory takes the place of the non-volatile storage medium, and the memory information is saved as the mirroring memory using the kdump. The mirroring memory can be read using the **export_kbox_img_to_txt** command.

7.3.3 Restarting the Kbox Service

This topic describes running environment, restrictions, and loading process of the kbox.

Prerequisites

- The kbox is running properly in the EulerOS.
- The dump medium is working properly.

Loading Process

In the EulerOS that is running normally, log in to the system as a user with the root rights. Ensure that the configuration file is modified based on the requirements.

In the shell command line environment, run the following command in any path:

```
#systemctl  restart kbox
```

Verifying Whether the Configuration File Is Successfully Loaded

NOTE

When the system is shut down and then reboot, the kbox service is not interrupted. The **systemctl stop kbox** command is called when the system is shut down and reboot. The kbox service is not interrupted. The interface is null.

Run the following command to view the running status of the kbox:

```
#systemct  status kbox
```

- If "active" is displayed, the kbox module is successfully loaded and running.
- If "active" is not displayed, the kbox module is not loaded or not successfully loaded. Check the kbox configuration file and parameters. Load the kbox module again after verification. For more information, see **7.5 Troubleshooting of Common Faults**.

7.3.4 Viewing Exception Information

This topic describes how to view exception information saved in the storage device.

Prerequisites

- The kbox is running properly in the EulerOS.
- The dump medium is working properly.
- Only the user with the root rights can view the kbox-related information.

Exporting Exception Information

Upon system boot, the kbox logs are saved in the path **/var/crash** (default location, you can change this location by setting the **path** parameter in the kdump configuration file). Logs for

each system exception are saved in the directory named by IP address-time point. The **xxx-xxx.img.gz** file in this directory is a kbox log.

NOTE

Currently, the kbox logs are saved in the directory named "127.0.0.1-time point". 127.0.0.1 indicates that the kbox logs are only saved in the current host and will not be sent to other servers.

```
[root@localhost 127.0.0.1-2015.11.24-12:08:11]# ls
100000000-1000000.img.gz  vmcore-dmesg.txt
```

Run the following command to decompress and parse the logs:

```
gunzip  100000000-1000000.img.gz
export_kbox_img_to_txt  -i 100000000-1000000.img
```

The system displays "export_kbox_img_to_txt: parse kbox data into /tmp/kbox_log.txt success.

The **kbox_log.txt** file is the parsed kbox log.

You can run the **vi** command to view the log that records the exception information. The OOM exception information is used as an example. For details, see **7.2.3 Providing System OOM Information**.

```
localhost:~ # vi /var/log/kbox/kbox_oom_1.txt
*****area type:oom - location in oom area:1*****
----------KBOX_START----------
oom time:19701209163422-a4a7
Current Pid: 1978, comm: bash Tainted: GF          O --------------   3.10.0-229
.20.1.23.x86_64 #1
Call Trace:
 [<ffffffffa024ca33>] kbox_dump_oom+0x13/0x30 [kbox]
 [<ffffffffa024bad7>] kbox_oom_callback+0x107/0x210 [kbox]
 [<ffffffff816101cc>] notifier_call_chain+0x4c/0x70
 [<ffffffff8109d63d>] __blocking_notifier_call_chain+0x4d/0x70
 [<ffffffff8109d676>] blocking_notifier_call_chain+0x16/0x20
 [<ffffffff8115bc5e>] out_of_memory+0x4e/0x4f0
 [<ffffffff8115bc10>] ? clear_zonelist_oom+0xa0/0xa0
 [<ffffffff815fe32f>] ? printk+0x77/0x8e
 [<ffffffff8115bc10>] ? clear_zonelist_oom+0xa0/0xa0
 [<ffffffffa0423a31>] dev_wr_actions+0x931/0x13a0 [kpgen_kbox]
 [<ffffffffa0424e06>] dev_wr_handler+0xa6/0x120 [kpgen_kbox]
 [<ffffffff811c7f0d>] vfs_write+0xbd/0x1e0
 [<ffffffff816100c3>] ? trace_do_page_fault+0x43/0x100
 [<ffffffff811c8958>] SyS_write+0x58/0xb0
 [<ffffffff816147c9>] system_call_fastpath+0x16/0x1b
oom stack:
 ffff8800ba60bc68 ffff8800ba60bcc0 ffff8800ba60bce4 0000000000000000
 ffff8800ba60be40 0000000000000000 ffff8800ba60bcc0 ffffffffa024c00a
 0000000000000002 000000004546aaf3 0000000000000000 ffff8800ba60bcd0
 ffffffffa024ca33 ffff8800ba60bd60 ffffffffa024bad7 3037393181610220
 3433363139303231 00376134612d3232 0000002e81eca3e0 ffff880036750c00
 ffffffff0000002f ffff88013b035800 0000000000000000 000000004546aaf3
 ffffffff819c2760 000000000000002f ffff8800ba60be20 0000000000000020
 000000004546aaf3 00000000ffffffff 0000000000000000 ffff8800ba60bd98
 ffffffff816101cc ffffffff8197bcc0 0000000000000000 0000000000000000
 ffff8800ba60be40 00000000ffffffff ffff8800ba60bdd8 ffffffff8109d63d
 0000000000000000 0000000000000000 0000000000000001 ffff8800ba60bf48
 ffff88013ffd9000 00000000000000d0 ffff8800ba60bde8 ffffffff8109d676
```

7.4 Viewing Kbox Information

This topic describes how to view the kbox-related information.

Prerequisites

- The kbox is running properly in the EulerOS.
- The dump medium is working properly.

Procedure

- Check the running status of the kbox. Run the following commands:

```
systemctl  status  kbox kbox.service - Crash message collector Loaded: loaded
(/usr/lib/systemd/system/kbox.service; disabled) Active: active (exited) since
Thu 2015-11-26 10:39:47 EST; 13s ago Process: 26545 ExecStart=/usr/sbin/kbox
start (code=exited, status=0/SUCCESS) Main PID: 26545 (code=exited, status=0/
SUCCESS)
 Nov 26 10:39:46 localhost.localdomain kbox[26545]: kbox: kbox module has been
loaded successfully! Nov 26 10:39:46 localhost.localdomain kbox[26545]: driver:
drivers have been loaded successfully! Nov 26 10:39:47 localhost.localdomain
systemd[1]: Started Crash message collector.
```

- View the output information.
  - If the active state is displayed, the kbox is running properly, as shown in the following figure.

```
kbox.service - Crash message collector
   Loaded: loaded (/usr/lib/systemd/system/kbox.service; disabled)
   Active: active (exited) since Thu 2015-11-26 10:39:47 EST; 13s ago
```

  - If the failed state is displayed, the kbox is not loaded to the system, as shown in the following figure.

```
kbox.service - Crash message collector
   Loaded: loaded (/usr/lib/systemd/system/kbox.service; disabled)
   Active: failed (Result: exit-code) since Thu 2015-11-26 10:44:43 EST; 5s ago
  Process: 29347 ExecStart=/usr/sbin/kbox start (code=exited, status=1/FAILURE)
 Main PID: 29347 (code=exited, status=1/FAILURE)
```

- View the temporary storage region of the system.

  After the kbox is started, by default, it creates multiple temporary storage regions to save the corresponding logs. Run the following command to view the existing storage regions in the system:

```
Storage:/ #kboxstatus ------------- product      info -------------
product       name: pangea hardware version: pangea-v2

 ------------- enabled    events ------------- panic_event    reboot_event
emerge_event die_event      oom_event      rlock_event

 ------------- enabled  storages ------------- bios


 the regions and storages list:

 ------------- regions      list ------------- console die emerge message oom
panic reboot rlock

 ------------- storages     list ------------- biosnvram
```

  Table 7-7 describes the temporary storage regions.

Table 7-7 Description of storage regions

Name	Description
console	Saves log information output to the console after the storage exception occurs. If excessive information is output, the earliest information will be overwritten. The size of this storage region is 32 KB.

Name	Description
die	Saves exception information captured when the kernel process encounters the OOPS (memory destroyed or using a null pointer) event. The size of this storage region is 128 KB.
message	Saves the latest 64 KB information output by the kernel when the memory exception occurs.
oom	Saves the information captured for the OOM event. The size of this storage region is 256 KB.

- View the global information of the storage device.

  View the storage devices registered in the system.

  The storage devices indicate the storage mediums registered in the kbox module, including the NVRAM, PCIE device memory, and BMC device memory. After an exception occurs, the kbox updates the log information saved in the temporary storage regions to these storage devices. Currently, the CloudEdge uses the HMEM (ordinary memory) as the storage device by default. An example is as follows:

```
localhost:~ # kboxstatus
the effective info in /etc/kbox/config:

------------ product      info ------------
product       name: euler
version     number: V200R002C10
frame       number: 1
slot        number: 0
locate        info: 0

------------ enabled    events ------------
panic_event    die_event      oom_event      oom_call_panic


------------ enabled  storages ------------
hmem


the regions and storages list:

------------ regions      list ------------
console die message oom panic

------------ storages     list ------------
hmem mem_info
localhost:~ #
```

7.5 Troubleshooting of Common Faults

This section describes common faults and corresponding handling methods of the kbox.

Kbox Failed to Export Logs

1. Run the **systemctl status kbox** command to check whether the kbox service is started. If not, run the **systemctl start kbox** command to start the kbox service. If the startup fails, check whether the kbox configuration is correct.
2. Run the **systemctl status kdump** command to check whether the kdump is normally started. If not, run the **systemctl start kdump** command to start the kdump service. If the startup fails, check whether the kdump configuration is correct.

NOTE

The kbox logs are saved by the kdump.

Kbox Failed to Capture Exception Information

Run the **systemctl status kbox** command to check whether the kbox service is started. If not, run the **systemctl start kbox** command to start the kbox service.

Kbox Service Failed to Start

Perform the following steps to identify the fault:

1. Run the **cat /etc/os-release** command to check whether the current OS is EulerOS. Currently, the kbox supports only the EulerOS and cannot be used in Linux systems in other versions.
2. Run the **systemctl status kbox** and **journalctl -l** commands to display the prompt information generated when the kbox service fails to start.
3. Run the **uname -m** command to check the hardware architecture. Check whether the host computer adopts the x86 architecture. Currently, the kbox supports only the x86 architecture.
4. Check whether the kbox configuration file is correct. Ensure that the modified configuration items are correct.
5. If the reserved memory is faulty, the kbox also fails to start.

   During the boot phase, the system reserves 16 MB memory space for kbox commissioning. To achieve memory reservation, add **reserve_kbox_mem=16M** to the **cmdline** parameter in the grub configuration file.

   The memory reservation algorithm has the following steps:

   a. In the e820 directed acyclic graph, the system traverse each memory segment in the usable state from the high memory address to the low memory address, until a segment that satisfies the 16 MB requirement is found.

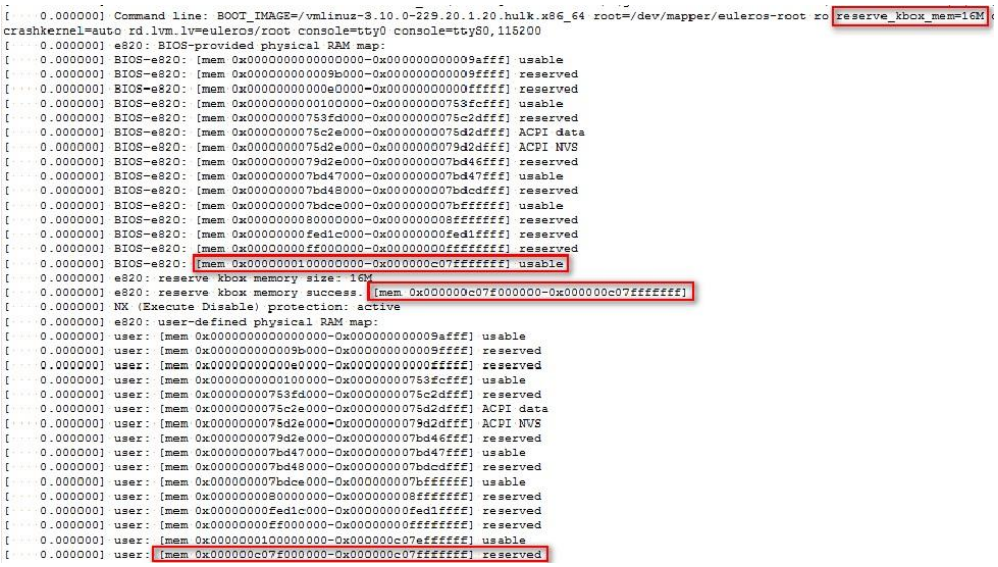
   b. Reserve 16 MB memory space in the found memory segment from the high memory address to the low memory address.

      Note:

      (1) If the available memory space of the system is smaller than 16 MB, the kbox fails to start.

      (2) When the memmap is used to reserve the memory space, ensure that it is not conflicted with the reserve_kbox_mem memory segment. The memmap does not identify the state of the memory segment when reserving the memory space. Therefore, the forcible allocation mode may cause a conflict with the reserved memory space.

The following figure shows the changes of the address segment range and memory state before and after the reserve_kbox_mem memory segment is reserved.



7.6 Appendix

7.6.1 Command References

NOTE

The **/usr/sbin/kbox** is an internal command used for starting the kbox service. The users must not use this command.

Starting the Kbox Service

systemctl start kbox

Restarting the Kbox Service

systemctl restart kbox

Viewing the Kbox State

systemctl status kbox

Viewing the Valid Configuration Information, Name of the Temporary Storage Region, and Registered Storage Devices

kboxstatus

NOTE

If the kbox service is not started, the current temporary storage regions and registered storage devices are not displayed.

Exporting Log Information

The **os_kbox_config** command is used to export the logs in the non-volatile storage devices.

NOTE

Because the CloudEdge is not equipped with any non-volatile storage device, this command does not have an actual effect in the current CloudEdge environment.

os_kbox_config -o dev=[Storage device name],type=[Type of exported logs],index=[SN of the log] | -h | -v

For details about the parameters, see **Table 7-8**.

Table 7-8 Parameter description

Parameter Name	Description
dev=[Storage device name]	Name of the storage device (BIOS NVRAM, HMEM, or PCI) registered in the current system, used to save exception information.
type=[Type of the exported log]	Type of the exported logs. It is the same as the region name and can be set to **oom/ console/message/die**.
index=[SN of the log]	The SN of the log. The value ranges from **1** to **32**. **1** indicates the latest log. Up to 32 logs can be recorded.
-h	View the help information.
-v	View the kbox version information.

Parsing Log Files

Because the general server is not equipped with the non-volatile storage system, the kbox saves logs only in the common memory. When the system crashes, the kdump saves the common memory as a mirroring memory file. The kbox parses the mirroring memory file using **export_kbox_img_to_txt** and outputs log files in the text format.

export_kbox_img_to_txt -i kbox file

NOTE

- By default, the kbox files are compressed using the gzip. Before parsing the files, run the **gunzip xxx-xxx.img.gz** command to decompress the files.

- This command is only used to parse the kbox log files on the general server.

8 kdump

Kernel Dump (kdump) is a crash dump mechanism that collects memory information in the event of a kernel crash. When triggered, kdump exports memory information (vmcores) that can be analyzed for determining the cause of the crash.

8.1 Overview

When the running kernel crashes, for example, because of the use of the magic key c, the operating system switches to the dump-capture kernel where memory information at the time of crash is collected and saved.

NOTE

By default, kdump does not save user data at the time of kernel crash. Exert caution when modifying kdump configurations as this may result in some of user data being saved. To stop kdump service, follow the guidelines provided in **8.4 Managing kdump Service**.

8.2 Constraints

- Only memory information of EulerOS can be dumped.
- Some of vmcores created by kdump may be incomplete, depending on the disk write speed, memory usage, and watchdog timer interval.
- The dump duration heavily relies on memory size.
- The following must be configured in the configuration file of the virtual machine (VM) that is being created. Otherwise, kdump does not work on the VM.

  <features>

  <acpi/>

  </features>

8.3 Configuring kdump Parameters

Getting kdump Information

NOTE

By default, kdump enters the working state after the operating system is booted.

The kdump configuration file /etc/kdump.conf contains the following common parameters:

Parameter	Description
default	Default action that will be taken in the event of kdump failure.
path	Path where kdump logs are saved. The path must be located on the root partition (/).
dracut_args	Passes kdump initrd parameters to dracut.

Configuring kdump Parameters

NOTICE

- Retain default settings in the kdump configuration file. Changes to kdump settings may lead to unexpected errors.

- The kernel command line (cmdline) must contain crashkernel=auto. To list parameters specified on the cmdline, run the cat /proc/cmdline command.

Table 8-1 describes key parameters in the kdump configuration file /etc/kdump.conf.

Table 8-1 Key kdump parameters

Parameter	Description	Value
default	Default action that will be taken in the event of kdump failure.	Default: reboot
path	Path where kdump logs are saved. **NOTE** The path must be located on the root partition (/).	Default: /var/crash
dracut_args	Passes kdump initrd parameters to dracut.	Null
core_collector	A tool for compressing kdump logs.	Default: makedumpfile with the -d option (filtering level) set to 31
kdump_obj	Indicator of whether to save kbox logs or vmcores.	If kbox is correctly configured, kbox logs are saved. Otherwise, vmcores are saved instead.
kbox_mem	Memory area where kbox logs are saved.	The memory area is automatically allocated by kbox. Do not attempt to change it.
keep_old_dumps	The number of historical kdump logs that will be saved.	-1: Only current logs are saved.

8.4 Managing kdump Service

- Command for stopping kdump service:
```
#systemctl  stop  kdump
```
- Command for starting kdump service:
```
#systemctl  start  kdump
```
- Command for restarting kdump service:
```
#systemctl  restart  kdump
```
- Command for querying kdump:
```
#systemctl  status  kdump
```

8.5 Parsing vmcores using the crash Tool

To parse vmcores using the crash tool provided by Euler linux, perform the following steps:

1. Find the crash installation package vmcore_debug_tool.tar.gz from the Software \02.Tools\DebugTools\ directory of VMP installation package.

2. Decompress vmcore_debug_tool.tar.gz to any directory on EulerOS 2.1. Copy the vmcores saved by kdump to that directory.
3. In that directory, run the following command to parse vmcores:

   ```
   ./crash vmlinux-3.4.24.15-0.11-default vmlinux-3.4.24.15-0.11-default.debug
   vmcore
   ```

   The parsed vmcores contains information about the kernel, CPU, memory, panic, and so on.

   ```
        KERNEL: vmlinux-3.4.24.15-0.11-default
     DEBUGINFO: vmlinux-3.4.24.15-0.11-default.debug
       DUMPFILE: vmcore
           CPUS: 4
           DATE: Mon Nov 24 15:35:44 2014
         UPTIME: 854015929139 days, 18:47:25
   LOAD AVERAGE: 5.99, 6.00, 6.04
          TASKS: 108
       NODENAME: Storage
        RELEASE: 3.4.24.15-0.11-default
        VERSION: #1 SMP Tue Nov 12 06:28:53 UTC 2013 (0c85715)
        MACHINE: x86_64  (1800 Mhz)
         MEMORY: 4 GB
          PANIC: ""
            PID: 60724
        COMMAND: "exhaustmem"
           TASK: ffff8800593cc500  [THREAD_INFO: ffff880062046000]
            CPU: 1
          STATE: TASK_RUNNING (PANIC)
   crash>
   ```

   To list the commands supported by the crash tool, run the help command. The following is an example output of the help command:

   ```
   crash> help
    *               files           mach            repeat          timer
   ascii           fuser           mount           search          vm
   bt              gdb             net             set             vtop
   btop            help            p               sig             waitq
   dev             ipcs            ps              struct
   whatisdis             irq             pte             swap            wr
   eval            kmem            ptob            sym             q
   exit            list            ptov            sys
    extend          log             rd              task
   ```

NOTE

The log command does not work if logs in the cyclic buffer are redirected to the nonvolatile random access memory (NVRAM). Information about memory segments allocated to NVRAM cannot be saved into vmcores.

To display details of a particular command, for example, the bt command, run the man bt or help bt command. Example details of the bt command

```
crash> man bt
 NAME
   bt - backtrace
 SYNOPSIS
   bt [-a|-g|-r|-t|-T|-l|-e|-E|-f|-F|-o|-O] [-R ref] [-I ip] [-S sp] [pid | task]
 DESCRIPTION
   Display a kernel stack backtrace.  If no arguments are given, the stack
   trace of the current context will be displayed.
        -a  displays the stack traces of the active task on each CPU.
            (only applicable to crash dumps)
        -g  displays the stack traces of all threads in the thread group of
            the target task; the thread group leader will be displayed first.
        -r  display raw stack data, consisting of a memory dump of the two
            pages of memory containing the task_union structure.
……
```

8.6 Common Troubleshooting

Illegal Stack Access

1. Compile code to simulate illegal stack access.

```
#include <linux/kernel.h>
 #include <linux/module.h>
 static int __init a_init(void)
 {
        char msg[10] = {0};
        printk("a init\n");
        strcpy(msg, "this modules is testing module,
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa");
        return 0;
 }
 static void __exit a_exit(void)
 {
        printk("a exit\n");
 }
 module_init(a_init)
 module_exit(a_exit)
 MODULE_LICENSE("GPL");
```

2. Perform the following diagnostic steps:

   Run the bt command to display stack traces of the current context. The stack traces show that the stack of a particular process cannot be parsed.

```
crash> bt
PID: 13360  TASK: ffff8801e386c140  CPU: 2   COMMAND: "insmod"
 #0 [ffff8801fddabc60] machine_kexec at ffffffff8101fc9b
 #1 [ffff8801fddabcb0] crash_kexec at ffffffff81082115
 #2 [ffff8801fddabd80] show_registers at ffffffff81006979
 #3 [ffff8801fddabde0] __die at ffffffff8138e3e7
 #4 [ffff8801fddabe10] die at ffffffff81007623
 #5 [ffff8801fddabe40] do_general_protection at ffffffff8138e0a9
 #6 [ffff8801fddabe70] general_protection at ffffffff8138d7cf
    [exception RIP: init_module+60]
    RIP: ffffffffa087e03c  RSP: ffff8801fddabf20  RFLAGS: 00010246
    RAX: 0000000000000000  RBX: 0000000000614010  RCX: 0000000000000000
    RDX: 0000000000000007  RSI: ffffffffa0a4c117  RDI: ffff8801fddabfe7
    RBP: 20676e6974736574   R8: 0000000000000000   R9: 0000000000000400
    R10: 0000000000000000  R11: 0000000000000000  R12: ffffffffa087e000
    R13: 0000000000000000  R14: 00007f139d2f6000  R15: 0000000000000002
    ORIG_RAX: ffffffffffffffff  CS: 0010  SS: 0018
    RIP: 6161616161616161  RSP: 00007fff37c795f8  RFLAGS: 00010202
    RAX: 6161616161616161  RBX: 6161616161616161  RCX: 6161616161616161
    RDX: 6161616161616161  RSI: 6161616161616161  RDI: 6161616161616161
    RBP: 6161616161616161   R8: 6161616161616161   R9: 6161616161616161
    R10: 6161616161616161  R11: 6161616161616161  R12: 6161616161616161
    R13: 6161616161616161  R14: 6161616161616161  R15: 6161616161616161
    ORIG_RAX: 6161616161616161  CS: 616161616161  SS: 002b
bt: WARNING: possibly bogus exception frame
```

   Determine the stack memory based on RSP: ffff8801fddabf20. Identify the location of error code in the stack memory.

```
crash> rd ffff8801fddabe00 100
ffff8801fddabe00:  0000000000000292 ffff8801fddabe38   ........8.......
ffff8801fddabe10:  ffffffff81007623 ffff8801e386c140   #v......@.......
ffff8801fddabe20:  ffff8801fddabe78 0000000000000000   x...............
ffff8801fddabe30:  00007f139d2f6000 ffff8801fddabe68   .`/.....h.......
ffff8801fddabe40:  ffffffff8138e0a9 ffffffff8156f610   ..8.......V.....
ffff8801fddabe50:  0000000000000001 ffffffffa087e000   ................
ffff8801fddabe60:  0000000000000000 20676e6974736574   ........testing
ffff8801fddabe70:  ffffffff8138d7cf 0000000000000002   ..8.............
ffff8801fddabe80:  00007f139d2f6000 0000000000000000   .`/.............
ffff8801fddabe90:  ffffffffa087e000 20676e6974736574   ........testing
ffff8801fddabea0:  0000000000614010 0000000000000000   .@a.............
ffff8801fddabeb0:  0000000000000000 0000000000000400   ................
ffff8801fddabec0:  0000000000000000 0000000000000000   ................
ffff8801fddabed0:  0000000000000000 0000000000000007   ................
ffff8801fddabee0:  ffffffffa0a4c117 ffff8801fddabfe7   ................
ffff8801fddabef0:  ffffffffffffffff ffffffffa087e03c   ........<.......
ffff8801fddabf00:  0000000000000010 0000000000010246   ........F.......
ffff8801fddabf10:  ffff8801fddabf20 0000000000000018    ...............
ffff8801fddabf20:  202c656c75646f6d 6161616161616161   module, aaaaaaaa
ffff8801fddabf30:  6161616161616161 6161616161616161   aaaaaaaaaaaaaaaa
ffff8801fddabf40:  6161616161616161 6161616161616161   aaaaaaaaaaaaaaaa
ffff8801fddabf50:  6161616161616161 6161616161616161   aaaaaaaaaaaaaaaa
ffff8801fddabf60:  6161616161616161 6161616161616161   aaaaaaaaaaaaaaaa
ffff8801fddabf70:  6161616161616161 6161616161616161   aaaaaaaaaaaaaaaa
ffff8801fddabf80:  6161616161616161 6161616161616161   aaaaaaaaaaaaaaaa
ffff8801fddabf90:  6161616161616161 6161616161616161   aaaaaaaaaaaaaaaa
ffff8801fddabfa0:  6161616161616161 6161616161616161   aaaaaaaaaaaaaaaa
ffff8801fddabfb0:  6161616161616161 6161616161616161   aaaaaaaaaaaaaaaa
ffff8801fddabfc0:  6161616161616161 6161616161616161   aaaaaaaaaaaaaaaa
ffff8801fddabfd0:  6161616161616161 6161616161616161   aaaaaaaaaaaaaaaa
ffff8801fddabfe0:  0000616161616161 0000000000010202   aaaaaa..........
ffff8801fddabff0:  00007fff37c795f8 000000000000002b   ...7....+.......
```

Illegal Access to Page Tables

1. Compile code to simulate illegal access to page tables.

```
#include <linux/kernel.h>
 #include <linux/module.h>
 #include <linux/kprobes.h>
 #include <linux/kallsyms.h>
 #include <linux/syscalls.h>
 #include <linux/slab.h>
 #include <linux/kdebug.h>
 #include <asm/apic.h>
 #include <asm/pgalloc.h>
 static int __init jprobe_init(void)
 {
         unsigned long address;
         pgd_t *pgd;
         pud_t *pud;
         pmd_t *pmd;
         printk(KERN_INFO "zk--- in \n");
         address = (unsigned long)kmalloc(512 ,GFP_KERNEL);
         pgd = pgd_offset(current->active_mm, address);
         pud = pud_offset(pgd, address);
         pmd = pmd_offset(pud, address);
         memset(pmd, 0, 16);
         printk("test: %x \n", *(int*)address);
         return 0;
 }
 static void __exit jprobe_exit(void)
 {
```

```
        printk(KERN_INFO "zk--- out \n");
}
module_init(jprobe_init)
module_exit(jprobe_exit)
MODULE_LICENSE("GPL");
```

2. Perform the following diagnostic steps:

   Analyze the erroneous stack and finds that it is due to an address error.

```
crash> bt
PID: 13016  TASK: ffff88003595c080  CPU: 0   COMMAND: "insmod"
 #0 [ffff88008b0f7b10] machine_kexec at ffffffff81021a54
 #1 [ffff88008b0f7b60] crash_kexec at ffffffff8108baf8
 #2 [ffff88008b0f7c30] show_registers at ffffffff81006bf8
 #3 [ffff88008b0f7c90] __die at ffffffff81400f2e
 #4 [ffff88008b0f7cc0] no_context at ffffffff8102f733
 #5 [ffff88008b0f7d00] __bad_area_nosemaphore at ffffffff8102f925
 #6 [ffff88008b0f7dd0] bad_area_nosemaphore at ffffffff8102f9fe
 #7 [ffff88008b0f7de0] do_page_fault at ffffffff81402b0e
 #8 [ffff88008b0f7e30] page_fault at ffffffff8140029f
    [exception RIP: acct_collect+88]
    RIP: ffffffff8108af68  RSP: ffff88008b0f7ee8  RFLAGS: 00010286
    RAX: ffff8800658ba6b8  RBX: 0000000000002000  RCX: ffff88006b050480
    RDX: 0000000000000000  RSI: 0000000000000015  RDI: ffffffff81732a20
    RBP: ffff88008b0f7f08   R8: 0000000000000000   R9: fffffffffffffa8
    R10: 00007f7c07212700  R11: 0000000000000246  R12: ffff8800346ac400
    R13: 0000000000000000  R14: 0000000000000000  R15: 00007f7c071d0010
    ORIG_RAX: ffffffffffffffff  CS: 0010  SS: 0018
 #9 [ffff88008b0f7f10] do_exit at ffffffff81052b11
#10 [ffff88008b0f7f40] do_group_exit at ffffffff81052c0d
#11 [ffff88008b0f7f70] sys_exit_group at ffffffff81052c92
#12 [ffff88008b0f7f80] system_call_fastpath at ffffffff81002ffb
    RIP: 00007f7c06d2c2a8  RSP: 00007fffd5e227a8  RFLAGS: 00010246
    RAX: 00000000000000e7  RBX: ffffffff81002ffb  RCX: 0000000000000000
    RDX: 0000000000000000  RSI: 000000000000003c  RDI: 0000000000000000
    RBP: 0000000000000000   R8: 00000000000000e7   R9: fffffffffffffa8
    R10: 00007f7c07212700  R11: 0000000000000246  R12: ffffffff81052c92
    R13: ffff88008b0f7f78  R14: 00007fffd5e237d4  R15: 0000000000000001
    ORIG_RAX: 00000000000000e7  CS: 0033  SS: 002b
```

   Analyze the instruction returned in response to RIP: ffffffff8108af68 and find that the instruction uses the (RAX+0x10) address.

```
crash> dis ffffffff8108af68
0xffffffff8108af68 <acct_collect+88>:   add    0x10(%rax),%rbx
```

   Analyze the page table that contains the address and find that the PMD of the page table incorrectly points to 0.

```
crash> vtop ffff8800658ba6c8
VIRTUAL           PHYSICAL
ffff8800658ba6c8  658ba6c8

PML4 DIRECTORY: ffffffff81a04000
PAGE DIRECTORY: 1a05063
   PUD: 1a05008 => 100067
   PMD: 100960 => 0

      PAGE        PHYSICAL      MAPPING       INDEX CNT FLAGS
ffffea0001962e80  658ba000                0        0  1 100000000000100
```

   Determine the suspected module based on the content in the memory that has been illegally accessed or based on the logs generated around the time of crash.

Overwriting of Static Variables

Run the sym -l command to list kernel symbols. Determine the addresses of the overwritten static variables based on the kernel symbols. Find the static variables near the addresses of the overwritten static variables. It is very likely that a particular variable overflows, causing neighboring variables to be overwritten.

```
ffffffff81a16100 (d) vm_table
ffffffff81a16d80 (d) fs_table
ffffffff81a17320 (d) debug_table
ffffffff81a173c0 (d) min_sched_granularity_ns
ffffffff81a173c4 (d) max_sched_granularity_ns
ffffffff81a173c8 (d) max_wakeup_granularity_ns
ffffffff81a173cc (d) min_sched_shares_ratelimit
ffffffff81a173d0 (d) max_sched_shares_ratelimit
ffffffff81a173d4 (d) max_sched_tunable_scaling
```

Illegal Memory Access in DMA

- Symptom

  There are no regular stack traces. All code areas are identical and their code is fe 0b ad ca.

  ```
  [ 3051.054204] Call Trace:
   [ 3051.057035]  [<ffffffff8115faa9>] path_openat+0xd9/0x420
   [ 3051.063054]  [<ffffffff8115ff2c>] do_filp_open+0x4c/0xc0
   [ 3051.069103]  [<ffffffff81150b71>] do_sys_open+0x171/0x1f0
   [ 3051.075223]  [<ffffffff8144fc53>] ia32_do_call+0x13/0x13
   [ 3051.081261] Code: fe 0b ad ca fe 0b ad ca fe 0b ad ca fe 0b ad ca fe 0b
  ad ca fe 0b ad ca fe 0b ad ca fe 0b ad ca fe 0b ad ca fe 0b ad ca fe 0b ad
  <ca> fe 0b ad ca fe 0b ad ca fe 0b ad ca fe 0b ad ca fe 0b ad ca
  ```

- Diagnosis

  The kernel code area is read-only and cannot be modified using a linear address. A physical address must have been used to modify the kernel code area. Both the direct memory access (DMA) controller and the basic input/output system (BIOS) are capable of operating on physical memory. Considering that the content stored in the target address is irrelevant to the DMA controller, then BIOS is suspected of causing the problem.

  BIOS code analysis discovers a driver code error. Due to the error, BIOS is overwritten during DMA and then illegally accesses kernel memory while BIOS is running.

Deadlock

- Diagnosis

  The crux of diagnosis lies in determining the stack being used by each CPU at the time of deadlock. To list all active stacks of the CPU, run the bt -a command. Then parse the data structure of lock to determine which thread was holding the lock.

- Common types of deadlock
  - A non-atomic procedure, such as sleep schedule, is executed in the spinlock protection area.
  - It takes long to execute logic in the spinlock protection area.
  - AB-BA deadlock
  - AA deadlock

- Ring lock in which modules in complex architecture wait for each other in a loop.
- Mismatch between lock and unlock, resulting in changes to lock pointer

9 Appendix

9.1 File Types and Names

In Linux systems, file types are usually inferred from file names.

Table 9-1 Compressed files

File Name	Description
.bz2	File compressed using the bzip2 compressor.
.gz	File compressed using the gzip compressor.
.tar	File compressed using the tar compressor.
.tbz	File compressed using the tar and bzip compressors.
.tgz	File compressed using the tar and gzip compressors.
.zip	File compressed using the ZIP compressor. Such files are generally used by MS-DOS applications.

Table 9-2 System files

File Name	Description
.conf or .cfg	Configuration file.
.lock	Lock file, which is used to determine whether a program or device is being used.
.rpm	Software package file.

Table 9-3 Programming and script files

File Name	Description
.c	Source code file in C language.
.cpp	Source code file in C++ language.
.h	Header file in C or C++ language.
.o	Program object file.
.pl	Perl script.
.py	Python script.
.so	Library file.
.sh	shell script.
.tcl	TCL script.

9.2 Common Directories

Solid knowledge of directory structure on the operating system helps you quickly find files during file operations and system management.

The file system uses a hierarchical, tree-like directory structure. The base of the file system hierarchy begins at the root (often referred to as the forward slash / since the full path to it is /). Directories branch off the root. The tree root and branches are directories or folders, whereas leaves are files.

Here are the most common top level directories.

- /bin: command files executable by common users.
- /sbin: commands executable by non-common users. Sometimes, the directory may also hold commands executable by common users.
- /lib: shared library files used by programs in /bin and /sbin directories.

- /usr: files and directories related to system users, as well as primary programs.

  The /usr hierarchy is as follows:

  - /usr/bin: most user commands
  - /usr/sbin: system commands
  - /usr/lib/: library files used by user programs
- /etc: system configuration files.
- /root: home directory for the system administrator (root or admin user).
- /home: user home directory in which user files, including configuration files, documents, and data, are located.
- /dev: device files. In Linux, devices are represented by files so that they can be operated in the same way as files are operated.
- /media: file system mount points, which are typically used for installing removable media, partitions of other file systems (for example, DOS), network file systems, or any installable file systems.
- /boot: files used for booting the kernel and operating system.
- /var: variable files, including spool directories and files, logging data, lock files, and temporary files.
- /proc: a virtual file system capable of storing memory images of the operating system without occupying disk space.
- /initrd: directory where initrd.img files are mounted at the system boot time and the required device modules are loaded.
- /opt: optional files and programs. The /opt directory is often used by third-party developers to install or uninstall software packages.
- /tmp: temporary files required by users and programs. Files in the /tmp directory are automatically deleted.
- /lost+found: files recovered in system repair.

9.3 Terms

account

A combination of name, directories, password, and shell of a login user.

alias

A mechanism that enables a replacement of a word in a shell command by another string. To list all defined aliases, type **alias** at the shell prompt.

ARP

Address Resolution Protocol (ARP), which is used to map IP addresses to MAC addresses of equipment in a local area network.

batch

A processing mode in which a processor executes a series of jobs without manual intervention until it is ready to receive another series of jobs.

boot

A process in which the operating system is loaded into memory after completion of power-on self-tests.

bootdisk

A removable digital data storage medium from which a computer can load and run (boot) an operating system or utility program.

BSD

Berkeley Software Distribution (BSD) is a Unix operating system derivative developed and distributed by the Computer Systems Research Group (CSRG) of the University of California, Berkeley.

buffer

A memory region with fixed capacity to load region mode files, system partition tables, processes that are being executed, and so on.

buffer cache

An embedded memory through which external devices read or write data from or into buffers.

CHAP

Challenge-Handshake Authentication Protocol (CHAP), a communication protocol used by ISP to verify its clients. Unlike the Password Authentication Protocol (PAP), CHAP periodically initiates verification after the initial verification.

client

A program or computer that connects temporarily to other programs or computers and issues instructive commands or information requests to the latter. It is a part of client/server system.

client/server system

A system architecture composed of a server and one or more clients.

compilation

The act of transforming source code written in a human-readable programming language such as the C language into binary files that can be recognized by machines.

completion

A feature of command line interpreters, in which shell automatically fills in partially typed commands.

compression

A method of encoding information using fewer characters or smaller files than the original representation. Common compressors include compress, zip, gzip, and bzip2.

console

A machine used by users to operate a super computer and often referred to as a terminal. The console to operate a PC is the keyboard and screen.

cookies

A small piece of data sent from a website and stored in the user's web browser while the user is browsing. Every time the user loads the website, the browser sends the cookie back to the server to notify the user's previous activity.

DHCP

Dynamic Host Configuration Protocol (DHCP) is a communication protocol used within a local area network to allocate IP addresses dynamically to DHCP clients.

DMA

Direct memory access (DMA), a feature of computer systems that allows interface devices to access main system memory independently of the central processing unit (CPU).

DNS

Domain Name System (DNS), a mechanism of mapping machine names to IP addresses recognizable for network devices, or conversely.

DPMS

Display Power Management System (DPMS), a standard for managing power supply of monitors. Example usage includes shutting off the monitor after a period of idle time to save power.

editor

A type of program used for editing plain text files. Popular GNU/Linux editors include Emacs and VIM.

email

A method of exchanging digital messages between users in the same network. The addresses of the sender and recipients must be correct.

9.4 Comparison Between Common Linux and DOS Commands

Table 9-4 Comparison between common Linux and DOS commands

Description	Linux Command	DOS Command
Copy a file	cp	copy
List files	ls	dir
Clear screen	clear	cls
Move a file	mv	move
Delete a file	rm	del
Create a directory	mkdir	mkdir
View files	less	more
Rename a file	mv	ren
View the current path	pwd	chdir
Display the output on your screen	echo	echo

Description	Linux Command	DOS Command
Find a string in a file	grep	find
Terminate a program	exit	exit
Display or set date	date	date
Display occupied memory	free	mem

