

**2024-2025 Huawei ICT  
Competition National Stage  
Computing Track  
Lab Test**



**Huawei Technologies Co., LTD.**

**All Rights Reserved.**



# Contents

---

<b>1 Background .....</b>	<b>3</b>
<b>2 About the Test.....</b>	<b>4</b>
2.1 Scoring Description .....	4
2.2 Test Considerations.....	4
2.3 Test Platform .....	5
2.4 Record Your Answers.....	5
<b>3 Exams .....</b>	<b>6</b>
3.1 openEuler Tasks.....	6
3.1.1 Background .....	6
3.1.2 Exam Environment.....	6
3.1.3 Tool Used in the Exam .....	6
3.1.4 Exam Requirements.....	6
3.1.5 Exam Content .....	7
3.2 openGauss Tasks .....	12
3.2.1 Background .....	12
3.2.2 Exam Environment.....	12
3.2.3 Tool Used in the Exam .....	12
3.2.4 Exam Requirements.....	12
3.2.5 Exam Content .....	12



# 1 Background

---

Education is one of the most important sectors to any country, with governments around the world investing heavily in education to promote a better future for the young generations. Similarly, every year more students are choosing to study information technology, and an indicator to determine competition among students is score management. As schools grow at scale, the workloads of student score management are more important than ever, but this can be hindered by slow, manual processing of mass data. This practical exam requires examinees to develop a student score management system on the openEuler OS and openGauss database.



# 2 About the Test

## 2.1 Scoring Description

This test consists of two sections - openEuler and openGauss - each accounting for 500 points. The former covers system monitoring, security, and tuning, and the latter covers data objects, SQL statements, and data export. The total score is 1000.

Section		Proportion	Score
openEuler	System monitoring	15%	150
	System security	20%	200
	System tuning	15%	150
openGauss	Data objects	15%	150
	SQL statements	25%	250
	Data export	10%	100

## 2.2 Test Considerations

- Read the *Exam Guide* and exam tasks before you begin.
- For tasks with multiple options, determine the most suitable choice to fit the requirements.
- The software used in the computing test is as follows:

Software	Description
PuTTY	Remote login tool
VirtualBox	Virtual machine software
openEuler	openEuler OS



## 2.3 Test Platform

Install PuTTY and VirtualBox on the PC and obtain the openEuler ISO image. For details, see the *Exam Guide*.

## 2.4 Record Your Answers

All answers must be recorded as **screenshots**. Read the *Exam Guide* for details.



# 3 Exams

## 3.1 openEuler Tasks

### 3.1.1 Background

Servers must provide long-term stable services for external systems over a secure and reliable environment. One issue is the lack of resources and performance bottlenecks caused by spikes in service requests. Therefore, system monitoring and tuning should be performed for timely analysis.

### 3.1.2 Exam Environment

Set up the environment on your PC by referring to the *Exam Guide*.

### 3.1.3 Tool Used in the Exam

The following table lists the tool used in this exam.

Software	Description
PuTTY	Remote login tool

### 3.1.4 Exam Requirements

1. All screenshots of key steps must be in PNG format and named according to the task.
2. Strictly follow the configuration and naming instructions. Any deviation will result in deducted points.
3. PuTTY must be used for logging in to the openEuler OS.



## 3.1.5 Exam Content

### 3.1.5.1 Exam Tasks

Each step in a task is scored separately. Manage your time for each question accordingly.

#### Task 1: System Monitoring (150 Points)

##### Subtask 1: Display process information. (10 points)

- a. Load the current CPU usage, total number of processes, and number of zombie processes.

##### Screenshots:

- a. Take a screenshot of the command output on the terminal and save the image as **1-1-1CPU**.

##### Subtask 2: Adjust process priority. (20 points)

- a. Determine the state and process ID of the sshd service.
- b. Increase the priority of the sshd service by 10 levels.

##### Screenshots:

- a. Take a screenshot of the state and process ID of the sshd service and save the image as **1-1-2sshd**.
- b. Take a screenshot of the command for modifying the process priority and save the image as **1-1-3level**.

##### Subtask 3: Schedule a task. (20 points)

- a. In the **/home** directory, compile a **hello.sh** script that displays **Hello, openEuler, have a nice day!**.
- b. Set the script task to run at 08:00 every day.

##### Screenshots:

- a. Take a screenshot of the command for scheduling tasks and save the image as **1-1-4command**
- b. Take a screenshot of the task scheduling content and save the image as **1-1-5task**.

##### Subtask 4: Set audit rules. (20 points)



- a. Use the **auditctl** tool to monitor the operations of modifying the permissions and attributes of the **/home/hello.sh** file. You can run the **auditctl -h** command to view the usage and parameters.

**Screenshots:**

- a. Take a screenshot of the command that modified the file permission and save the image as **1-1-6audit**.

**Subtask 5: Change file permissions. (20 points)**

- a. Grant the execute permission on the **/home/hello.sh** file to the owner.

**Screenshots:**

- a. Take a screenshot of the modified file permissions and save the image as **1-1-7 permission**.

**Subtask 6: Change the file owner. (20 points)**

- a. Change the owner of the **/home/hello.sh** file to **omm**.

**Screenshots:**

- a. Take a screenshot of the command for changing the owner of **hello.sh** and save the image as **1-1-8omm**.

**Subtask 7: Run command logs. (20 points)**

- a. Use the **ausearch** tool to view the audit logs of **/etc/passwd** in a human-readable format. You can run the **ausearch -h** command to view the usage and parameters.

**Screenshots:**

- a. Take a screenshot of the command on the terminal and save the image as **1-1-9ausearch**.

**Subtask 8: Filter processes. (20 points)**

- a. Use the **ps** command to view the top three processes of memory usage.

**Screenshots:**

- a. Take a screenshot of the command and its output on the terminal and save the image as **1-1-10ps**.



## Task 2: System Security (200 Points)

### Subtask 1: Modify configuration files to enforce user password (60 points)

- Minimum length of a user password: 8
- A user password must contain at least: one lowercase letter, one uppercase letter, one number, and one special character
- Maximum validity period of user passwords: 90 days

#### Screenshots:

- Log in as user **test**, change the password to one with 7 or fewer characters. Take a screenshot and save the image as **1-2-1length**.
- Log in as user **test**, change a password to one that does not meet the complexity requirements. Take a screenshot and save the image as **1-2-2complex**.
- Use the **chage -l { user }** command to view the maximum validity period of the **test** user's password. Take a screenshot of the output and save the image as **1-2-3time**.

### Subtask 2: User permissions (50 points)

- Disable login for user **test** by setting its shell to **/sbin/nologin**.
- Enable user **test** again and add it to the sudo group.

#### Screenshots:

- View the content of the **/etc/passwd** file, take a screenshot of the line where user **test** is located, and then save the image as **1-2-4info**.
- Switch to user **test**, run a file creation command with **sudo** (for example, **sudo touch /etc/test.txt**), and then check whether the file is created. Take a screenshot of the command and its result and save the image as **1-2-5sudo**.

### Subtask 3: Minimize firewall policies. (60 points)

- Disable all ingress ports except HTTP ports 80 and 443.
- Open port 22 for SSH connection from specified servers (for example, the local host machine).

#### Screenshots:

- Run the **iptables -L** command to view all firewall policies, then take a screenshot of the command output and save the image as **1-2-6iptables**.
- Take a screenshot of the VM connection using SSH on the local host machine before and after enabling the default SSH port, and save the image as **1-2-7ssh**.

**Subtask 4: Prevent log deletion. (30 points)**

- a. Allow only edit operations on system log files (for example, `/var/log/secure`), and forbid delete operations.

**Screenshots:**

- a. Run the `lsattr {log file}` command to view the log file attributes. Run the `rm {log file}` command, take a screenshot, and save the image as **1-2-8lsattr**.

**Task 3: System Tuning (150 Points)****Subtask 1: View memory information. (20 points)**

- a. Run the `free` command to view system memory information. The command output must contain the following information:
  - a) Total memory (MB)
  - b) Used memory (MB)

**Screenshots:**

- a. Take a screenshot of the command and its output and enclose the preceding information in a red box. Save the image as **1-3-1free**.

**Subtask 2: Optimize memory. (40 points)**

- a. Use the `sysctl` command to display the `swappiness` parameter of the system.
- b. Use the `sysctl` command to adjust the `swappiness` value by 10 so that the swap space is preferred over memory, so as to reduce memory usage.

**Screenshots:**

- a. Take a screenshot of the command to display the `swappiness` parameter and its output. Save the image as **1-3-2swappiness**.
- b. Take a screenshot of the command to modify the `swappiness` parameter and the modified `swappiness` value. Save the image as **1-3-3parameter**.

**Subtask 3: Query drive I/O information. (30 points)**

- a. Use the `iostat` command to view drive I/O data, including the following information:
  - a) Average wait time for each I/O request
  - b) Average I/O queue length
  - c) Percentage of I/O operation time in a period

**Screenshots:**



- a. Take a screenshot of the command and its output and enclose the preceding information in a red box. Save the image as **1-3-4io**.

#### Subtask 4: Optimize drives. (60 points)

- a. Configure the server to prevent the number of opened files exceeding the threshold when the server is busy.
  - a) Use the **ulimit** command to view the maximum number of opened files in the system.
  - b) Use the **ulimit** command to temporarily double the threshold.
- b. **dirty\_background\_ratio** specifies the percentage of memory that can be occupied by dirty data. When the percentage is reached, data is written to drives.
  - a) Use the **sysctl** command to check the **dirty\_background\_ratio** parameter of the system.
  - b) Use the **sysctl** command to adjust the value of **dirty\_background\_ratio** by five to allow the memory to store more dirty data, so as to reduce I/O operations.

#### Screenshots:

- a. Take a screenshot of the command to view maximum number of opened files in the system, then save its output. Save the image as **1-3-5num**.

Take a screenshot of the command to modify the maximum number of opened files in the system and the modified value. Save the image as **1-3-6rate**.
- b. Take a screenshot of the command to view the **dirty\_background\_ratio** parameter and its output. Save the image as **1-3-7ratio**.

Take a screenshot of the command to view the **dirty\_background\_ratio** parameter and the modified value. Save the image as **1-3-8ratio2**.



## 3.2 openGauss Tasks

### 3.2.1 Background

This section comprises three parts: 1. Create database objects, including tablespaces, databases, users, and tables; 2. Write SQL statements in the **student** database; and 3. Use the `gs_dump` tool to export data.

### 3.2.2 Exam Environment

Set up the environment on your PC by referring to the *Exam Guide*.

### 3.2.3 Tool Used in the Exam

Software	Description
PuTTY	Remote login tool

### 3.2.4 Exam Requirements

1. All screenshots of key steps must be in PNG format and named according to the task.
2. Strictly follow the configuration and naming instructions. Any deviation will result in deducted points.
3. PuTTY must be used for logging in to the openEuler OS.

### 3.2.5 Exam Content

#### 3.2.5.1 Exam Tasks

**Each step in a task is scored separately. Manage your time for each question accordingly.**

#### Task 1: Creating Data Objects (150 Points)

Run the following command to log in to the postgres database.

```
[root@openEuler ~]# su - omm
[omm@openEuler ~]$ gsql -d postgres -p 15432 -r
gsql ((openGauss 3.0.0 build 02c14696) compiled at 2022-04-01 18:12:19 commit 0 last
mr )
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.

openGauss=#
```

#### Subtask 1: Create a tablespace named `tbs`. (20 points)



- a. Create the **tbs** tablespace in **/home/omm/tbs**.

**Screenshots:**

- a. Take a screenshot of the command and save the image as **2-1-1createTBS**.

**Subtask 2: Create a database named mydb. (30 points)**

- a. Create the **mydb** database.
- b. Set the tablespace of the **mydb** database to **tbs**.

**Screenshots:**

- a. Take a screenshot of the command to create the **mydb** database and save the image as **2-1-2createDB**.
- b. Take a screenshot of the command to set the default tablespace to **tbs** and save the image as **2-1-3alterDB**.

**Subtask 3: Create a database user. (30 points)**

- a. Create database user **test** and set the password to **openGauss@123**.

**Screenshots:**

- a. Take a screenshot of the command and save the image as **2-1-4createUser**.

**Subtask 4: Set user privileges. (20 points)**

- a. Grant all privileges of the **mydb** database to the user **test**.

**Screenshots:**

- a. Take a screenshot of the command and save the image as **2-1-5userPrivileges**.

**Subtask 5: Create a data table. (50 points)**

- a. Create a data table **tb1**. The first column **id** is of the integer type, the second column **name** is of the character type (max. 20 characters), and the third column **date** is of the date type.
- b. In the table, input a data record whose ID is **1**, name is **James**, and date is **1992-10-01**

**Screenshots:**

- a. Take a screenshot of the command to create the table and save the image as **2-1-6createTable**.
- b. Take a screenshot of the command to insert data and save the image as **2-1-7insert**.



## Task 2: Writing SQL Statements (250 Points)

The **student** database consists of four tables: **student**, **teacher**, **course**, and **score**. The following lists the table names and column names in the tables. Write SQL statements based on the following subtask requirements.

Table	Column
student	<b>sid</b> : student ID; <b>sname</b> : student name; <b>sage</b> : date of birth; <b>ssex</b> : gender
course	<b>cid</b> : course ID; <b>cname</b> : course name; <b>tid</b> : teacher ID
teacher	<b>tid</b> : teacher ID; <b>tname</b> : teacher name
score	<b>sid</b> : student ID; <b>cid</b> : course ID; <b>score</b> : score

Log in to the **student** database.

```
[root@openEuler ~]# su - omm
[omm@openEuler ~]$ gsql -d student -p 15432 -r
gsql ((openGauss 3.0.0 build 02c14696) compiled at 2022-04-01 18:12:19 commit 0 last
mr )
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.

student=#
```

### Subtask 1: Display table data. (30 points)

- Query information about Kenny in the **student** table.

**Screenshots:**

- Take a screenshot of the command and output, and save the image as **2-2-1kenny**.

### Subtask 2: Use aggregate functions to display table data. (30 points)

- Query the number of students in the **student** table.

**Screenshots:**

- Take a screenshot of the command and output, and save the image as **2-2-2num**.

### Subtask 3: Use group statements to display table data. (30 points)

- Query the numbers of males and females in the **student** table.

**Screenshots:**

- Take a screenshot of the command and output, and save the image as **2-2-3gender**.



**Subtask 4: Use conditional statements to display table data. (30 points)**

- a. Query information about students born in 2012 in the **student** table. Note: Use the `to_char()` function to convert **sbirthday** in the **student** table to a character type.

**Screenshots:**

- a. Take a screenshot of the command and output, and save the image as **2-2-4char**.

**Subtask 5: Join and query multiple tables (1). (30 points)**

- a. Query the SID and total score of Megan in the **student** table.

**Screenshots:**

- a. Take a screenshot of the command and output, and save the image as **2-2-5sid**.

**Subtask 6: Join and query multiple tables (2). (50 points)**

- a. Query students whose total scores are greater than 190 and display their names and total scores in descending order.

**Screenshots:**

- a. Take a screenshot of the command and output, and save the image as **2-2-6total**.

**Subtask 7: Join and query multiple tables (3). (50 points)**

- a. Query the names and scores of students whose Math score is higher than 87.

**Screenshots:**

- a. Take a screenshot of the command and output, and save the image as **2-2-7math**.

**Task 3: Exporting Data (100 Points)**

**Subtask 1: Export data using gs\_dump. (50 points)**

- a. Use `gs_dump` to export the **student** database. The file path is **/home/omm/backup** and the file name is **student\_all.sql**.

**Screenshots:**

- a. Take a screenshot of the command, the exported file name, and file size, and save the image as **2-3-1student**.

Note: Switch to user **omm** and use `gs_dump` to export data. Run **gs\_dump --help** to view related parameters, the **cd** command to go to the **/home/omm/backup** directory, and the **ls -l** command to display the command output.



**Subtask 2: Export data using gs\_dump. (50 points)**

- a. Use `gs_dump` to export the **score** table from the **student** database. The file path is **/home/omm/backup** and the file name is **student\_score.sql**.

**Screenshots:**

- a. Take a screenshot of the command, the exported file name, and file size, and save the image as **2-3-2score**.

Note: Switch to user **omm** and use `gs_dump` to export data. Run `gs_dump --help` to view related parameters, the `cd` command to go to the **/home/omm/backup** directory, and the `ls -l` command to display the command output.