HUAWEI

Huawei ICT Competition 2025–2026 Exam Outline

Practice Competition – Computing Track

I. C. The Future

01 Overview

Competition Stage	Exam Type	Duration	Number of Questions	Question Type	Total Score	Number of Contestants	Remarks
Preliminary Stage	Written	60 minutes	60	True or false, single-answer, and multiple-answer questions	1000	Individual	From January 1, 2024 to the end date of the Preliminary Stage, 50 bonus points will be acquired for passing any of HCIA-openEuler/openGauss/Kunpeng, 100 bonus points for any of HCIP-openEuler/openGauss/Kunpeng, and 200 bonus points for any of HCIE-openEuler/openGauss. These bonus points can be combined up to a maximum of 200 points. **Note:** The Uniportal account used for the competition registration must be the same as that used for the certification. Otherwise, no bonus point can be given.
National Stage	Written	120 minutes	150	True or false, single-answer, and multiple-answer questions	1000	Individual	/
Regional Stage	Lab	4 hours	/	Comprehensive lab	1000	3 (as a team)	In Regional Stage, each of the three contestants in a team needs to complete the test questions for the written exam, and they will together complete the tasks for the lab exam. One team can submit only one set of answers for each of their written and lab exams. Total score = 30% x Written exam score + 70% x Comprehensive lab exam score
Global Stage	Lab	8 hours	/	Comprehensive lab	1000		/

02 Weighting

Competition Stage Technical Direction	Preliminary Stage	National Stage	Regional Stage	Global Stage
openEuler	50%	50%	50%	50%
openGauss	30%	30%	30%	30%
Kunpeng Application Developer	20%	20%	20%	20%

03 Scope

3.1. Exam Content Overview

The Computing Track exam will test your knowledge on openEuler, openGauss, and Kunpeng Application Developer, including but not limited to openEuler development history, basic operations, memory, processes, and file systems; openGauss overview, connections, access, databases (fully-encrypted and tamper-proof), cluster management, monitoring, and O&M; Kunpeng architecture and solution, application development, application porting, and performance tuning.

3.2. Exam Scope

Direction	Category	Key Items	Description	Preliminary Stage	National Stage	Regional Stage	Global Stage
				HCIA	HCIP	HCIE	HCIE or above
openEuler	Basic principles	Basic concepts	Basic concepts, main features, and development history of openEuler	√	√	√	√
		Architecture	Architecture of the Kunpeng processor	√	√	√	√
	openEuler basics	openEuler installation	openEuler installation and login methods	√	√	√	√
		Command line interface basics	Bash shell usage and common operations	√	√	√	√
		Vim editor basics	Usage of the Vi and Vim editors	√	√	√	√
		Shell script basics	Basics of shell scripting and programming	√	√	√	√
	Management	Memory	Paging (mechanisms, management, tables, MMU principles, address translation) and memory allocation differences between malloc, kmalloc, and vmalloc with their use cases	√	√	√	√
		Processes	Address space layout, system calls (usage, principles, and their relationship with library functions), scheduling (concepts and algorithms), synchronization, and inter-process communication	√	√	√	√
		Permissions	Users, user groups, file permissions, and other permissions	√	√	√	√
		Software and services	Software packages, software installation using DNF or source code, and systemd	√	√	√	√
		Networks	Common network models and protocols	√	√	√	√
		File systems and storage	Basic file system concepts, drive storage mounting and usage, and logical volume management	√	√	√	√
		System	Management of tasks, network connections, and processes	√	√	√	√
	Security	Security management	Read/Write/Execute (rwx) permissions, firewall, and SELinux policies		√	√	√
		Security hardening	Basic concepts and usage of secGear			√	√
	Performance optimization	Performance monitoring	System performance monitoring tools and analysis of system metrics (CPU, memory, drive I/O, and network)	√	√	√	√
		Performance tuning	Common performance tuning methods and A-Tune concepts and usage			√	√
		Compilation optimization	Common compilation optimization methods and Plug-IN (PIN) framework		√	√	√
	Business service management and configuration	Apache	Basic installation and configuration		√	√	√
		Nginx	Basic installation and configuration		√	√	√
		DNS	Working principles of the Domain Name System (DNS) and DNS server configuration		√	√	√
		MySQL	User addition and data query		√	√	√
		LNMP/LAMP	Joint configuration of software components		√	√	√
	Cluster software configuration	LVS	Installation, NAT mode, and direct routing configuration		√	√	√
		Nginx	Configuration of reverse proxy and load balancing		√	√	√
		HAProxy	Basic installation, configuration, and access control lists (ACLs)		√	√	√
		Keepalived	Basic installation and configuration		√	√	√
	Shared storage configuration	iSCSI	Installation, configuration, and mounting of iSCSI targets and initiators		√	√	√
		NFS	Installation, permission configuration, and automatic mounting of NFS		√	√	√
		GlusterFS	Volume types, high availability (HA) configuration, and automatic mounting		√	√	√
	Management automation	Ansible	Basic module functions and playbook creation		√	√	√
		SaltStack	Remote control and task orchestration		√	√	√
	Key features	Virtualization	Basic concepts and usage of QEMU and StratoVirt			√	√
		Containers	Basic concepts, usage, and container image building of Docker and iSulad			√	√
		Kubernetes	Kubernetes basics			√	√
		OpenStack	OpenStack basics			√	√
		CI/CD	Continuous integration and development based on Jenkins			√	√
		Porting	Basic concepts and usage of x2openEuler			√	√
		O&M	Basic concepts and usage of A-Ops, kernel live upgrade, and application hot patch			√	√
	Ecosystem	openEuler community	Community organizations, contribution, learning, and code release	√	√	√	√
openGauss	openGauss overview	Basic concepts	Basic openGauss concepts and functions	√	√	√	√
		Architecture	openGauss architecture (logical structure and physical architecture), main features, and components	√	√	√	√
	Installation and deployment	Installation and deployment	openGauss single-instance installation and deployment, primary/standby HA deployment, upgrade, and uninstallation	√	√	√	√
	Database management	Database and object management	Tablespace creation and management, users and roles, system catalogs and system views, high-risk data operations, partitioned table enhancement, and object management tools	√	√	√	√
		Import and export	Data import and export	√	√	√	√
		Connection and access	pg_hba/SSL and remote access, password control policies, terminal tools, and development tools for connecting to a database	√	√	√	√
		Routine O&M	Routine O&M, failover, cluster management components, and flashback		√	√	√
	Storage engines	Data storage structures	Row store, column store, and storage planning			√	√
		Tablespace management	Default tablespace and tablespace creation and management		√	√	√
		Partition management	Partition creation, deletion, merging, splitting, and exchanging		√	√	√
		Log management	System logs, performance logs, pg_xlog, and audit logs			√	√
	SQL engines	SQL basics	SQL syntax classification (DDL, DML, and DCL), common functions and operators, data dictionaries (system catalogs and system views), and data types	√		√	√
		Advanced SQL	Advanced SQL syntax (subquery, nested query, union query, and aggregate query), VACUUM operation, and compatibility plug-ins			√	√
		SQL execution plans	Execution operators (table join, table scan, table aggregation, and set operations), EXPLAIN usage, and execution modes			√	√
	Database development	API development	Database development specifications, JDBC-based development, ODBC-based development, and Python-based development			√	√
		Connection and access	Development tools, middleware, and programming languages for connecting to a database		√	√	√
	Stored procedures and triggers	Stored procedures	Declaration syntax, basic statements, dynamic statements, control statements, and cursors		√	√	√
		Triggers	Triggers creation, modification, and deletion		√	√	√
	Security management	Access control	Database connection control, SSL connection control, remote connection control, and connection authentication	√	√	√	√
		User management	Roles and users, user creation, modification, and deletion, and account security policies	√	√	√	√
		Permission management	Role-based permission management model and separation of duties		√	√	√
		Object management	Authorization operations and minimum authorization		√	√	√
		Data encryption	Row-level security policy, data masking (column-level access control), function encryption, transmission encryption, and transparent data encryption		√	√	√
		Security audit	Audit policies, audit enabling/disabling, and audit logs		√	√	√
	Migration tools	openGauss migration tools	One-stop migration, full migration, incremental migration, reverse migration, and data verification tools			√	√
	Performance tuning	OS parameter tuning	Memory, network, I/O, and CPU tuning			√	√
		Database system tuning	Vectorized executor, SMP, and Ustore tuning; key parameter tuning, and performance diagnosis and analysis			√	√
		SQL tuning	Slow SQL diagnosis, similar operator selection, plan hints, index optimization, table structure optimization, subquery optimization, and SQL statement rewriting			√	√
	Advanced features	Advanced security features	Fully-encrypted and tamper-proof databases			√	√
		MOT	Feature value, key technologies, management usage, and application scenarios			√	√
		AI features	AI4DB (index recommendation) and DB4AI			√	√
	Ecosystem	openGauss community	Community organizations, contribution, learning, and code release	√	√	√	√
Kunpeng	Kunpeng products	Basics	Computer system architecture, program running principles, architecture differences, and programming language differences	√	√	√	√
		Hardware & Software basics	Kunpeng processors, motherboards, and servers, openEuler, openGauss, and openLooKeng	√	√	√	√
	DevKit	Software porting	Porting principles, workflow, and policies, application packaging methods, and interpreted language code porting methods	√	√	√	√
			Common C/C++ porting and troubleshooting methods	√	√	√	√
			Common Fortran and Rust porting and troubleshooting methods	√	√	√	√
			Functions, installation, and deployment of Porting Advisor	√	√	√	√
		Compilers	Compilation principles, Compiler and Debugger, BiSheng Compiler, BiSheng JDK, and GCC for openEuler		√	√	√
		Performance profilers	Java Profiler, System Profiler, Tuning Assistant, and System Diagnosis	√	√	√	√
	BoostKit	Performance tuning	Performance tuning methodology and common analysis tools	√	√	√	√
			CPU, memory, drive I/O subsystem, and network subsystem tuning	√	√	√	√
			Common tuning methods and Java application tuning	√	√	√	√
		BoostKit for Big Data	Common big data component installation, deployment, tuning, and main features, including machine learning algorithms, graph analysis algorithms, and OmniRuntime		√	√	√
		BoostKit for Database	Common database component installation, deployment, tuning, and main features, including MySQL parallel optimization, MySQL lock-free optimization, NUMA scheduling optimization, and MySQL thread pool		√	√	√
		BoostKit for Virtualization	Installation, deployment, and optimization of QEMU, OpenStack, Kubernetes, and Docker		√	√	√
		BoostKit for HPC	HPC solution architecture, Donau Scheduler, Hyper MPI, and HPC software deployment and tuning		√	√	√
	Ecosystem	Kunpeng community	Community organizations, modules and their functions, contribution, learning, and code release	√	√	√	√

04 Note

This Exam Outline is for general use only, and does not cover all exam details.

Huawei Technologies Co., Ltd.   Huawei ICT Competition Organizing Committee