



Huawei ICT Competition 2023-2024

Exam Outline

Practice Competition - Computing Track

1. Overview

Competition Stage	Exam Type	Duration	Number of Questions	Question Type	Total Score	Number of Contestants	Remarks
Preliminary stage (mandatory)	Written	90 minutes	60	True or false, single-answer, and multiple-answer	1000	1	From January 1, 2023 to the specified end date, 50 bonus points are available for every HCIA-openEuler, HCIA-openGauss, and HCIA-Kunpeng certification, and 100 bonus points are available for every HCIP-openEuler, HCIP-openGauss, and HCIP-Kunpeng certification (maximum 200 points). The Uniportal account used for registration must be the same as that used for the certification. Otherwise, no bonus point can be given.
National stage (optional)	Written	90 minutes	90	True or false, single-answer, and multiple-answer	1000		
Regional stage	Written	60 minutes	60	True or false, single-answer, and multiple-answer	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 Average written exam score of the team + 70% x Comprehensive lab exam score.
	Lab	4 hours	/	Comprehensive lab	1000		
Global stage	Lab	8 hours	/	Comprehensive lab	1000		

2. Weighting

Direction	Stage	Preliminary Stage	National Stage	Regional Stage	Global Stage
openEuler		50%	50%	50%	50%
openGauss		50%	50%	30%	30%
Kunpeng Application Development		/	/	20%	20%

3. Scope

3.1. Overview of Exam Contents

The Computing Track exam contents cover openEuler, openGauss, and Kunpeng application development, 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 optimization.

3.2 Knowledge to Be Tested

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	Shell script usage, shell programming basics, and shell programming practices	√	√	√	√	
	Management	Memory	Memory	Paging mechanism, management of physical and virtual pages, working principles of page tables and MMUs, conversion between virtual and physical addresses, and differences between malloc, kmalloc, and vmalloc and their application scenarios	√	√	√	√
			Processes	Process address space layout, system call usage and principles, relationship between system calls and library functions, process scheduling concepts and algorithms, process 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 the systemd service	√	√	√	√	
		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) permission mechanism, firewall, and SELinux policies		√	√	√	
		Security hardening	Basic concepts and usage of secGear			√	√	
	Performance optimization	Performance monitoring	Usage of system performance monitoring tools and analysis of system metrics (such as CPU, memory, drive I/O, and network)	√	√	√	√	
		Performance tuning	Common performance optimization methods and basic concepts and usage of the A-Tune automatic tuning tool			√	√	
		Compilation optimization	Common compilation optimization methods and plug-in 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	Adding users and querying data		√	√	√	
	Cluster software configuration	LNMP/LAMP	Joint configuration of software components		√	√	√	
		LVS	Installation, NAT mode, and direct routing configuration		√	√	√	
		Nginx	Configuration of reverse proxy and load balancing		√	√	√	
		HAProxy	Basic installation and configuration and access control list (ACL)		√	√	√	
	Shared storage configuration	Keepalived	Basic installation and configuration		√	√	√	
		iSCSI	Installation, configuration, and mounting of iSCSI targets and initiators		√	√	√	
		NFS	Installation, permission configuration, and automatic mounting of NFS		√	√	√	
	Management automation	GlusterFS	Volume type, high availability (HA) configuration, and automatic mounting		√	√	√	
		Ansible	Basic module functions and playbook creation		√	√	√	
	Key features	Salt	Remote control and task orchestration		√	√	√	
		Virtualization	Basic concepts and usage of QEMU and StratoVirt			√	√	
		Container	Basic concepts and usage of Docker and ISulad			√	√	
		Kubernetes	Kubernetes basics			√	√	
		OpenStack	OpenStack basics			√	√	
		Porting	Basic concepts and usage of x2openEuler			√	√	
	Ecosystem	O&M	Basic concepts and usage of the A-Ops intelligent O&M tool, kernel live upgrade, and application hot patch			√	√	
		Community ecosystem	Community organizations, contribution, learning, and code release	√	√	√	√	
	openGauss	openGauss Overview	Basic concepts	Basic concepts, theories, and functions of openGauss	√	√	√	√
			Architecture	Logical and physical architectures, main features, and components of openGauss	√	√	√	√
		Database management	Installation and deployment	Single-instance installation and deployment, primary/standby HA deployment, upgrade, and uninstallation of openGauss	√	√	√	√
			Database and object management	Tablespace creation and management, users and roles, system catalogs and views, data import and export, high-risk 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, development tool connections	√	√	√	√
		Storage engine	Routine O&M	Routine O&M management, common fault switchover, cluster management components, and flashback		√	√	√
			Data storage structure	Row storage, column storage, and storage planning		√	√	√
			Tablespace management	Default tablespace and creation and management of tablespaces		√	√	√
			Partition management	Creation, deletion, merging, splitting, and exchange of partitions		√	√	√
		SQL engine	Log management	System logs, performance logs, pg_xlog, and audit logs		√	√	√
SQL basics			SQL language categories (DDL, DML, and DCL), common functions and operators, data dictionaries (system catalogs and views), and data types	√	√	√	√	
Advanced SQL			Advanced SQL syntax (subquery, nested query, union query, and aggregate query), vacuum operation, and compatibility plug-ins		√	√	√	
Database development		SQL execution plan	Execution of operators (table join, table scan, table aggregation, and set operations), usage of the EXPLAIN statement, and execution modes		√	√	√	
		Interface development	Database development specifications and development based on JDBC, ODBC, and Python interfaces			√	√	
Stored procedures and triggers		Connection and access	Development tools, middleware, and database connections based on programming languages		√	√	√	
		Stored procedures	DECLARE syntax, basic statements, dynamic statements, control statements, and cursors		√	√	√	
Security management		Triggers	Creation, modification, and deletion of triggers		√	√	√	
		Access control	Control and authentication of database, SSL, and remote connections	√	√	√	√	
		User management	Roles and users, creation, modification, and deletion of users, and account security policies	√	√	√	√	
	Permission management	Role-based permission management and separation of duties		√	√	√		
	Object management	Permission granting and least privileges		√	√	√		
	Data encryption	Row-level access control, data masking (column-level access control), and function, transmission, and transparent encryption		√	√	√		
Migration tools	Security audit	Policies, switches, and logs		√	√	√		
	Migration tools	Full, incremental, and reverse migration, and data check			√	√		
Advanced features	Performance tuning	openGauss performance tuning			√	√		
	Advanced security features	Slow SQL diagnosis, key parameter tuning, performance diagnosis and analysis, and SQL optimization			√	√		
	MOT	Fully-encrypted database and tamper-proof database			√	√		
Ecosystem	AI features	Benefits, key technologies, usage management, and application scenarios of MOT			√	√		
	Community ecosystem	Index recommendation (AI4DB) and DB4AI			√	√		
Kunpeng	Kunpeng products	Community ecosystem	Community organizations, contribution, learning, and code release	√	√	√	√	
		Hardware	Kunpeng processor, server motherboard, and server			√	√	
	DevKit	Software	openEuler, openGauss, and openLoongArch			√	√	
		Basics	Computer system architecture, program running principles, architecture differences, and programming language differences			√	√	
		Software porting	Porting principles, workflow, and policies, application packaging methods, and interpreted language code porting methods			√	√	
			Common C/C++ porting problem handling			√	√	
			Common Fortran and Rust porting problem handling			√	√	
		Compilers	Functions, installation, and deployment of Porting Advisor			√	√	
	Performance tuning	Compilers	Compilation principles, compilation and debugging tools, BiSheng compiler, BiSheng JDK, and GCC for openEuler			√	√	
		Hyper Tuner	Java performance analysis, system performance analysis, Tuning Assistant, and system diagnosis			√	√	
	BoostKit	Performance optimization methodology and common analysis tools	Performance optimization methodology and common analysis tools			√	√	
		CPU, memory, drive I/O subsystem, and network subsystem tuning	CPU, memory, drive I/O subsystem, and network subsystem tuning			√	√	
		Common tuning methods and Java application tuning	Common tuning methods and Java application tuning			√	√	
		Big data solution	Common big data component installation, deployment, tuning, and main features, including MySQL parallel algorithms, graph algorithms, and OmniRuntime			√	√	
	Database solution	Common database component installation, deployment, tuning, and main features, including MySQL parallel optimization, MySQL lock-free optimization, NUMA scheduling optimization, and MySQL thread pool	Common database component installation, deployment, tuning, and main features, including MySQL parallel optimization, MySQL lock-free optimization, NUMA scheduling optimization, and MySQL thread pool			√	√	
Installation, deployment, and optimization of QEMU, OpenStack, Kubernetes, and Docker		Installation, deployment, and optimization of QEMU, OpenStack, Kubernetes, and Docker			√	√		
HPC solution		HPC solution architecture, Donau Scheduler, HyperMPI, and HPC software deployment and tuning			√	√		
Ecosystem	Kunpeng community	Community organizations, modules and their functions, contribution, learning, and code release			√	√		
	Kunpeng ecosystem	Kunpeng computing industry, ecosystem strategy, intelligent base, Kunpeng OpenMind Initiative, internship program, developer program, and developer competition			√	√		

Note

This Exam Outline is for reference only. It does not cover all exam details.