



Hewlett Packard Enterprise

Course Datasheet

Linux Performance Tuning And Analysis (GL325)

Education Services course product number – H2UW1S

Course length – 4 days

Delivery mode – Instructor Led Training (ILT)
virtual Instructor Led Training (vILT)

Register – [Click here to go to HPE Learning Portal](#)

This class meets the needs of advanced Linux Systems Administrators expanding their knowledge of core workings of Linux. Theories and methodologies are covered for tuning, monitoring performance, and analyzing the performance of the Linux operating system.

Prerequisites

An intermediate knowledge of the Linux operating system is required. The equivalent knowledge can be found in the [U8583S](#) "Linux Fundamentals" and [H7091S](#) "Enterprise Linux Systems Administration" courses.

Course objectives

- Learn principles and methods of system tuning.
- Understand Linux architecture.
- Use system and networking benchmark tools for understanding system capabilities.
- Diagnose complex performance and functional system problems.
- Learn to profile system hardware and applications.
- Analyze system performance to identify patterns.
- Learn methods of remote system monitoring.

Supported Distributions

Red Hat Enterprise Linux 7

Detailed Course Outline

- **Tuning Principles & the Use Method**
 - What is Performance Tuning?
 - Performance Tuning Methodology
 - Model System

- Establish a Baseline
- Establish Performance Targets
- Identify Bottleneck
- Tune
- Measure, Monitor, Evaluate, Repeat?
- Final Thoughts
- The USE Method
- Gathering USE Metrics

- **Lab Tasks**
 - Determining the System's Configuration
 - USE Method: CPU
 - USE Method: Memory
 - USE Method: Network
 - USE Method: Storage I/O & Capacity

- **Linux Internals Review**
 - The Linux Operating System Components
 - Open Standards
 - Memory Layout
 - Important Kernel Threads
 - Device Files in Linux
 - Managing Linux Device Files
 - udev
 - Random Numbers and /dev/random
 - Block Devices and the Device Mapper
 - List Block Devices
 - SCSI Devices
 - File Systems with Linux
 - Hardware Discovery Tools
 - Cache and Write Barriers
 - Advanced LVM: Components & Object Tags
 - Advanced LVM: Automated Storage Tiering
 - Advanced LVM: Thin Provisioning
 - Advanced LVM: Striping & Mirroring
 - Advanced LVM: RAID Volumes
 - Kernel Modules

- **Lab Tasks**
 - Adjusting Kernel Options
 - Linux Kernel Driver Compilation

- **Linux Userspace Architecture**
 - systemd System and Service Manager
 - Using systemd
 - Security Model Overview

- **Lab Tasks**
 - Managing Services With Systemd's systemctl
 - Setting Limits with the pam_limits Modules
 - Operation of the setuid() and capset() System Calls

- **Log File Administration**
 - System Logging
 - systemd Journal
 - systemd Journal's journalctl
 - Secure Logging with Journal's Log Sealing
 - gnome-system-log
 - Rsyslog

- /etc/rsyslog.conf
- Log Management
- Log Anomaly Detector
- Sending logs from the shell

- **Lab Tasks**
 - Using the systemd Journal
 - Setting up a Full Debug Logfile

- **Monitoring & Troubleshooting**
 - Viewing Processes
 - System Status – Memory
 - System Status – I/O
 - System Status – CPU
 - Performance Trending with sar
 - Determining Service to Process Mapping
 - Analyzing Process Memory Usage with pmap
 - Realtime Monitoring of Resources — Cgroups
 - strace and ltrace
 - lsof and fuser
 - ss/netstat and rpcinfo
 - tcpdump and wireshark

- **Lab Tasks**
 - Process Related Tools
 - System Activity Reporter
 - Cgroup for Processes
 - Analyzing Memory Usage

- **BENCHMARK TOOLS**
 - Benchmark Tools
 - Disk/Filesystem Benchmarks
 - Network Benchmarks
 - Phoronix Test Suite

- **Lab Tasks**
 - Benchmarking Disk and Filesystem
 - Bonnie++ Visualization

- **SystemTap**
 - SystemTap Overview
 - SystemTap Features
 - SystemTap Requirements
 - Invoking SystemTap
 - The 5 passes of stap
 - SystemTap Probe Point Syntax
 - SystemTap Probe Modes
 - SystemTap Probe Points
 - SystemTap process Probes
 - Listing and Using Probe Points
 - Built-in SystemTap Functionality
 - SystemTap Example Repositories
 - SystemTap Advanced Example
 - SystemTap Documentation

- **Lab Tasks**
 - System Tap Intro
 - System Tap
 - Creating SystemTap Scripts

- SystemTap One-liners & Histograms
- **Performance Co-Pilot**
 - Performance Co-Pilot Introduction
 - PCP Architecture
 - PCP Performance Metrics
 - PCP Installation
 - PCP Performance Monitoring
 - PCP Centralized Logging
 - PCP Inference Engine
 - PCP Deployment Strategies
 - PCP Advanced Topics
- **Lab Tasks**
 - Performance Co-Pilot Introduction
 - Live Monitoring with pmchart
 - Install Web Monitoring Tools
- **Remote Metric Gathering With SNMP**
 - SNMP Overview
 - SNMP Operation - Query
 - MIBs and OID Namespace
 - SNMP Security
 - SNMP for Linux
- **Lab Tasks**
 - SNMP
- **Linux perf**
 - perf Overview
 - perf Commands Typical Workflow
 - perf Events
 - perf stat
 - perf record
 - perf report
- **Lab Tasks**
 - Perf
- **Linux Tunables**
 - Operating System Tuning Caveats
 - Consider OS (or Kernel) Upgrade
 - Tuning CPU - NUMA
 - Tuning CPU - Scheduling
 - Huge Memory Pages
 - Tuning Virtual Memory Manager
 - Tuning Virtual Memory Manager (cont.)
 - Dropping Caches
 - I/O Scheduler
 - Tuning Filesystems
 - Tuning Filesystems (cont.)
 - Network Tuning
 - tuned