Solaris 10 Operating System Internals

Duration: 5 Days

What you will learn

The Solaris 10 Operating System Internals course provides students with information about the various kernel subsystems, routines, and structures that make up the Solaris 10 Operating System. Students will use Solaris Dynamic Tracing (DTrace) to step through process creation, execution, signal delivery, and scheduling, correlating observations with source code available through OpenSolaris. The labs make extensive use of dtrace, kmdb, and mdb commands to examine the system structures on live systems. The labs also make use of OpenSolaris web access to facilitate understanding how the operating system works. The kernel subsystems covered include the multithreaded architecture, virtual memory, scheduling, process lifetime, signal management, the vnode layer, and file systems such as UFS, ZFS and swapfs.
Students who can benefit from this course include:
Programmers, System Engineers, Advanced System Administrators, and Support Personnel

Related Training

Required Prerequisites

Understand and be able to explain the concept of pointers, structures, unions, link lists, hashing, and binary trees

Ability to read and write scripts

Read C programs and explain the meaning of a = (struct foo *) b and int func(int)

Suggested Prerequisites
Manage system processes

Course Objectives

Explain step-by-step how a lock is acquired

Discuss the reason for priority inheritance and its implementation

Identify the steps performed in a virtual to physical memory address translation

List the process structures and routines needed to implement a scheduling class

List the process structures used to implement multiple scheduling classes and the fields in the time-sharing and real-time dispatch parameter tables

Describe the paging and swapping algorithms that manage physical memory as a cache

Describe process creation, execution, and termination

Discuss kernel thread scheduling and preemption

Use kmdb, mdb, and DTrace to locate and display the system structures for an open file in a given process

Describe the placement policies that the UNIX file system (UFS) uses to place inodes and blocks of data

Describe the disk layout of a ZFS file system

Describe the block allocation algorithm, and free space mapping for ZFS

Describe the ZIO pipeline

Course Topics

Introducing the Solaris 10 Operating System
Define the purpose of the operating system and explain the concept of kernel layering
Explain and diagram the segments that make up the process address space
Explain the trap mechanism
Differentiate between hardware and software interrupts
List the new features in recent releases of the Solaris 10 OS
Start using tools such as mdb, kmdb, and DTrace to examine kernel data structures

Examining the Multithread Architecture
Describe the kernal thread
Use the lightweight process (LWP)
List the structures that describe the state of a kernel thread, an LWP, and a process
Describe Sleep Queues
Explain how a Mutex lock works
Describe how a counting semaphore is implemented
Explain how a multiple-reader, single-writer lock works

Hardware Memory Management
Describe the steps in virtual-to-physical address translation
List the differences between the x86/x64 memory management unit (MMU) and the SPARC SFMMU
Describe types of cache implemented on Sun systems

Examining Software Memory Management
List the layers of the SunOS 5.10 software virtual memory (VM) system
List the mapping structures that make up process address space
Locate the page structures and process address space structures
Explain how the memory mapping and memory control system calls can be used to manage process memory

Examining Paging and Swapping
Explain the layered approach to page-fault handling
List the conditions under which the page daemon runs
List the functions of the page daemon
List the conditions under which the swapper runs

The swapfs File System
Describe the memory scructures used to implement the swapfs file system

List the advantages obtained by adding the swapfs file system

Scheduling

List at least two major barriers to real-time processing
Explain the difference between a fully preemptible kernel and a kernel with preemption points
List a routine used to place a thread on a dispatch queue
Describe when a thread is placed at the head of a dispatch queue
Describe how the sleep queues are ordered
Define a user-level and kernel-level preemption
Define deterministic dispatch latency
Define priority inversion

Process Lifetime

Explain the differences among the system calls used to create a new process
Describe the kernel routines used to implement process creation
List the different types of executables supported in the Solaris 10 OS
Explain the routines used to implement executable and linking format (ELF) executables
List the advantages of the ELF executable format
Describe the actions taken by a process when it exits

Signals

List the different types of signals that can be delivered to a process or thread
Explain the difference between a trap signal and an interrupt signal
List the signal management routines and describe their functions
Describe what the signal facility is for, and how a signal is delivered

File Systems

Describe the vnode interface layer to a file system
List the four fields in a directory entry
Explain the advantages of ZFS
Describe the function of the superblock and cylinder group structures
List the fields in the disk inode structure and explain how they are used
Name the routines involved in determining the global placement policies
Describe how the ARC works