

UNIT 1 -- An Introduction to the Operating System	
What is an operating system?	1
Services Provided by the Operating System	3
Illusions	3
The System Call	5
UNIX Software component layers	5
About man pages	6
The UNIX I/O model	7
Standard I/O streams	8
Error reporting	9
Reading and Writing	10
Just a bunch of bytes	11
The stdio library	12
Buffering and the stdio library	12
UNIT 2 -- An Introduction to the File Systems	
The File System	1
What the File System provides	2
Flat file systems	2
UNIX Filesystems	3
Directories and inodes	5
Pathnames and Wildcards	7
Hard Links	7
Directories and link counts	12
Reading directories	13
Symbolic Links	14
The stat system call	17
Inode Types	19
The UNIX file permissions model	20
Mounted Volumes	22
Is that filesystem, or file system, or filesystem?	27
A review of filesystem-related system calls seen thus far	27
UNIT 3 -- Processes: fork,exec,exit,wait. I/O redirection	
The UNIX Process	1
Process State	1
The Virtual Address Space of a Process	1
Installing a new program with exec	3
Exec system call	4
Executing via an interpreter	5
The Environment	5
Starting a new process with fork	6
Clone	7
The file table and file descriptors	7
Dup and I/O redirection	9
fork and the file descriptor table	10
Typical shell I/O redirection	10
Expected file descriptor environment	11
Process termination	11
Typical fork/exec flow cycle	15
UNIT 4 -- Signals and Pipes	
Signals	1
kill system call	1
Pre-defined UNIX signal numbers and behaviors	2
Pending signals and signal masks	3
Signal delivery and disposition	3
Establishing signal disposition	4
The signal handler	4
setjmp/longjmp	5
Masking/Blocking signals	6
The sigaction interface	8
Real-time signals	9
Interrupted System Calls	9
SIGCHLD	10
Signal interactions with fork and exec	10
Pipes	11
Pipe properties	11
Flow Control	12
Atomicity	12
Connection Control	12

Named Pipes	13
UNIT 5 -- Virtual Memory	
Virtual Memory	1
Multi-Level Page Tables	3
Translations Cache	5
The UNIX user-level memory model	7
Demand-Paging	7
Paging-out and Backing Store	8
Paging-in, minor and major faults	9
Page / Protection Faults and signals	9
Mmap	10
Exec and virtual memory	12
Why the MAP_DENYWRITE flag?	13
Fork and copy-on-write	14
Appendix: How an executable file is created	16
UNIT 7 -- Concurrency and Synchronization	
Synchronization among Concurrent tasks	1
Shared Update Problem: A Simple Race Condition	2
Atomicity	4
Critical Regions and Mutual Exclusion	4
Implementing MUTEX by Disabling Interrupts or Signals	4
Spin Lock Mutex	5
Atomic Instructions	6
Code-based or Data-based?	7
Mutex locking with blocking/yielding	8
The deadlock problem	9
Deadlock Avoidance	9
Deadlock Detection	10
Live Lock	10
Fine vs Coarse Grain Locking	10
Read/write locking	11
Example of reader/writer locks	13
The problem of inconsistent views	13
Seqlocks	14
Semaphores	16
Condition Variables	16
Condition Variable Operations	17
Producer/Consumer problem using condition Variables	18
Some Golden Rules of Condition Variables	20
UNIT 8 -- Introduction to the Kernel	
What is the Kernel?	1
Boot-up	2
Re-entering the kernel	4
Exception Types	4
Interrupt Types	5
Important Shared Processor/Kernel Data Structures	5
Interrupt and Exception handling in hardware	6
Kernel control flow paths	8
Making a System Call	9
The 32-bit X86 API	9
Kernel-mode stack and the thread_info structure	12
The X86-64 API	13
Now, back to our system call, already in progress	14
System Call Return Value	15
Kernel system call hand-off to C	15
Returning from a system call	17
Deferred return from system call	17
Appendix - X86 Architecture and Assembly Language	18
Intel vs UNIX assembly syntax	18
Assembler Directives / Pseudo Opcodes	19
X86 Register Model	19
Segmentation, Special-Purpose and Additional Registers	20
Addressing Modes	20
Function Calling Convention	21
X86-64 Function Calling	23
X86-64 Global Variables	24
Caller/Callee saves	24
X86 General-Purpose Register Summary	25

UNIT 9 -- Task Switch / Sleep and Wakeup	
A Task Switch	1
Preemptive and Cooperative Multitasking	2
Process/task states	3
Data structures for tracking processes / tasks in Linux kernel	4
The struct task_struct	4
Sidebar: linked lists	6
Sleep and wakeup	7
Wait Queues	8
Exclusive vs non-exclusive waiting	9
Wait interrupted by signal	10
Wait queue example	11
Waking Up	12
A system call with blocking	14
Sleep/Wakeup Summary	19
The schedule() function	19
Kernel Mode Pre-emption	20
What schedule() does	20
Process and Thread Creation	25
Multiprocessor Considerations	32
Processos Termination	33
UNIT 10 -- The Kernel's Scheduler	
Scheduling	1
The UNIX model of priority	1
Real-Time Tasks	2
Quantum	2
The Old Linux Scheduler vs the CFS scheduler	2
Ideal and practical latency	3
Weighted timeslice	3
Virtual Runtime	4
Scheduler interactions with fork	5
The Scheduler Tick	5
Interactive Performance	7
Overview of the old Linux scheduler	8
Run Queues	8
Priority formula	10
Quantum	10
Active and Expired	11
Idle	11
scheduler_tick	11
Inserting a task into the run queue at wakeup	12
Scheduler interactions with fork	12
UNIT 11 -- Signals (Kernel Implementation)	
Signals	1
Signal data structures	1
Generating a signal	3
Following a kill	3
Signal Delivery	6
Setting up the stack frame to handle signal	10
sigsuspend	12
Restarting system calls	12
UNIT 12 -- Kernel Memory Management	
The Memory Management System	1
The Page Frame Pool	1
Addressing Issues on X86	1
Page Descriptor Table	2
Buddy System	3
Slab Allocator	4
Managing Virtual Address Space	5
Virtual Addressing on the X86	5
Representing the process address space	7
Address space release (munmap)	10
Copy-On-Write and Fork	11
Page Fault Handling	11
Stacks	15
Invalid Addresses and SIGSEGV	15
Invalid addresses in Kernel Mode / The fixup table	16
Benign Page Faults / Paging-in	16
Paging In Logic	17

On-demand creation of Page Tables	19
Paging-in from where?	20
Copy-On-Write (COW)	21
Page Frame Reclamation	22
Reverse Mapping	24
Anonymous Mappings	24
Swap Space and anonymous page-outs	25
File Mappings	26
Resources Exhaustion	29
Thrashing / Swap Token	29
Memory over-commit and the out-of-memory killer	29