Quizlet

Operating System, Question Bank Answers #1

Study online at quizlet.com/_3xr2we

1. **3-letter numbering scheme used in Linux.**	the most significant number defines the version number, if the middle number after first period is even it is deferred as stable version, if it is odd then called as development kernel. The last numbers after second period defines a release number and if it is odd development and if it is even stable release number.
2. **3 types of supported users in Linux**	owner - owner of file or app, group - groups that own file or app, others - all other users who are not owner and not in group bu has acess to system. chmod command used to change access rights
3. **All steps in System Startup**	1. After reset asserted, CS and IP registers are set with fixed value 0xfffffff0 that maps the address of BIOS program in ROM. 2. Processor executes from that address. 3. BIOS starts its execution and executes 4 tasks 4. BIOS runs Power-On-Self-Test, which checks whether all hardware devices connected to system is working properly 5. BIOS initialize all hardware devices 6. BIOS searches for operating system to boot 7. If found valid device, copies contents of first sector from hard disk (master boot record) into RAM, starting from physical address 0x00007c00, jumps to that address and executes. 8. In that address OS boot loader is executed. Boot loader loads in 2 stage as program is to big to fit into mbr in hard disk. It is loaded starting from 0x00007c00, moves itself to 0x00096a00, sets real mode stack (from 0x00098000 to 0x000969ff) loads second part of boot loader starting from 0x00096c00. Program reads map of available operating systems from disk and prompts user to choose OS. Then either boot loader copies first boot sector of corresponding OS to RAM, or directly copy kernel image into RAM. 9. Assume that kernel image must be booted. Boot loader invoke BIOS procedure to display "Loading" message. 10. Boot loader put initial portion of kernal image 512 bytes starting from 0x00090000 11. Then puts setup() function starting from 0x00090200 and setup() executes 12. setup() function switches cpu from real mode to protected mode and set up provisional layots for ram physical structure. 13. jumps to first startup_32() function which decompresses kernel image, invokes second startup_32() function 14. second startup_32() (process 0 or swapper)function initialize gdt, idt tables and values of gdtr and idtr, initialize segment registers with their final values, fills bss segment of the kernel with zeros. set up provisionals page tables, put global page directory address into cr3 register and enables paging by setting PG in the cr0 register, invokes start_kernel() (process 1 - init) which completes initialization of Linux 15. at the end login prompt appears.
4. **at least 4 types of shell on unix**	bourne shell, korn shell, c shell, bash shell, tee c shell
5. **BIOS bootstrap procedure**	1. BIOS runs Power-On-Self-Test, which checks whether all hardware devices connected to system is working properly 2. BIOS initialize all hardware devices 3. BIOS searches for operating system to boot 4. If found valid device, copies contents of first sector from hard disk (master boot record) into RAM, starting from physical address 0x00007c00, jumps to that address and executes.
6. **Boot Loader operations**	1. In that address OS boot loader is executed. Boot loader loads in 2 stage as program is to big to fit into mbr in hard disk. It is loaded starting from 0x00007c00, moves itself to 0x00096a00, sets real mode stack (from 0x00098000 to 0x000969ff) loads second part of boot loader starting from 0x00096c00. Program reads map of available operating systems from disk and prompts user to choose OS. Then either boot loader copies first boot sector of corresponding OS to RAM, or directly copy kernel image into RAM. 2. Assume that kernel image must be booted. Boot loader invoke BIOS procedure to display "Loading" message. 3. Boot loader put initial portion of kernal image 512 bytes starting from 0x00090000 4. Then puts setup() function starting from 0x00090200 and setup() executes
7. **bootstrapping**	This is the procedure to bring at least a portion of software into main memory and having the processor execute it.
8. **Define a Signal**	mechanism for notifying process about system event

9. **Define Kernel Control path.**	Sequence of instructions that should be executed to handle system call, exception, and interrupts
10. **Define Login Session.**	first command shell proccess created for the user
11. **Define Re-entrant function.**	functions which modify only local variables and does not alter global data structures.
12. **Define the following : a) File b) Directory**	File information container structured as sequence of bytes. Directory is array of links and each link maps filename to a file.
13. **Describe Re-entrant kernels**	Several processes run in kernel mode. Every process in kernel mode acts on its own set of memory locations and does not interfere with others
14. **Describe the different ways in which kernel routines can be activated in linux kernel.**	process invokes system call, cpu executing process signals exception, external device issues interrupt to CPU, kernel thread is executed;
15. **Describe the seven file types in Linux/Unix**	Regular File Directory Block-oriented device file character-oriented device file Sybmolic Links Socket Pipe and named pipe
16. **Describe the UNIX I/O primitives (System Calls) to manipulate Unix Files. open(), close(), write(), lseek(), read(), unlink(), rename()**	Describe the following functions to perform file operations. Indicate the correct format by writing all the parameters and their importance. a) open b) read c) write d) lseek e) close e) rename f) unlink int open(char * file_name, int flags, mode t_mode) return new fileDesciptor if OK, otherwise -1 on error file_name -> name of the file, flags -> that indicates how process intends to access the file, t_mode -> for access rights ssize_t read(int file_descriptor, void * buf, size_t n) file_descriptor -> opened file descriptor number, buf -> buffer to save contents of the file after reading, n -> how many bytes to read. ssize_t write(int file_descriptor, void * buf, size_t n) file_descriptor -> opened file descriptor number, buf -> buffer from which write contents of it to the file, n -> how many bytes to write. long lseek(int fd, long offset, int origin) file_descriptor -> opened file descriptor number, offset -> to set the current position offset for starting reading or writing, origin -> to measure the position to start int close(int fd) rename(char **oldpath, char** newpath) unlink(char * filename)
17. **... developed the self - contained Linux kernel from scratch with the major design goal of UNIX Compatibility.**	Linus Torvalds
18. **Discuss interleaving of Kernel Control paths by considering three different events.**	first when kernel control path verifies that request can not be satisfied immediately, CPU interleaves kernel control path; secondly, when CPU executing kernel control path detects an exception - for example page fault, CPU suspends kernel control path and starts execution of suitable procedure;thirdly, while CPU executing kernel control path, Cpu is interrupted bu external hardware
19. **Discuss Memory Area Descriptors.**	kernel usually stores a process virtual address space as a list of memory area descriptors

20. **Discuss the following :** **a) Swap Area** **b) hard disk cache**	a) Swap Area in order to extend the size of the virtual address space usable by the processes, Unix OS makes use of swap areas on disk b) hard disk cache Physical memory is used as a cache for hard diska and other block devices.
21. **Discuss the four solutions suggested for tackling Synchronization problem in Inter Process communication. Also indicate their merits and demerits.** **a)Nonpreemptive kernel;** **b) Interrupt Disabling;** **c) Semaphores** **d) Spin Locks**	Nonpreemptive kernels are good for uniprocessing systems, because when one process runs in kernel mode other process can wait until it stops, but it is not good for multiprocessing, because in multiprocessor system several processes can execute in kernel mode and interfere the shared data, where data inconsistency may occur. Interrupt disabling can be used in uniprocessor system as when interrupt disabled, no other external device can issue an interrupt. But in multiprocessor system it is totally not applicable, as there is no guarantee that no other CPU can interfere other shared data. Also when interrupt disabled, if executing process takes more time to finish, other external devices can end up in waiting state for a long time Semaphores are really good solution for uniprocessor systems, as every process before entering the critical region checks the semaphore value whether it is equal or more than 0. But it is not the best choice for multiprocessor system as time taken for updating semaphore could more expensive. Spin locks are best solution for multiprocessor systems where as another process that wants to interfere shared data of running process should spin until first process completes critical region. But it is useless for uniprocessor environment
22. **Distinguish between absolute pathname and relative pathname**	absolute path always starts with / which is deferred as root directory. relative path starts from current directory.
23. **Distinguish between foreground and Background processes.**	when a process has access to terminal, that process will be in foreground, if process has no access to terminal and does not interact with terminal
24. **Distinguish between Hard link and Soft link.**	when soft link deleted only link to that path is deleted, but if hard link deleted then the path itself is deleted. soft link can be created to both files and directories and among other filesystems, whereas hard links can be created only for files in the same filesystem
25. **Distinguish between Kernel mode and User mode**	Kernel mode gives priority to use all the resources of the system, User mode gives only limited access to files or directories created by that owner.
26. **Distinguish between Signals and Interrupts**	interrupt is asynchoronous event and communication between CPU and OS Kernel, while signal is synchronous event and communication between OS Kernel and OS processes. Interrupts may be initiated by the CPU. Signals may be initiated by the OS Kernel
27. **The dmesg command**	kernel log messages
28. **Explain Copy-On-Write (COW) approach.**	This approach used in fork creation, when fork creates child process, child process gets the pointer to read the code address space of its parent, and if child or parent writes into page or starts using some data, then new page is allocated for child process
29. **Explain deadlock using an example.**	when process p1 gains access to data structure a and process p2 gains access to access to data structure b, but then p1 waits for access to b and p2 waits access to a
30. **Explain Device Driver.**	composition of data structures and functions that control one or more devices, such as hard disks, keyboards, mouses, monitors and other devices connected to SCSI bus.
31. **Explain Kernel Memory Allocator (KMA)**	Subsystem that tries to satisfy the requests for memory areas from all parts of the system
32. **Explain OS types**	Batch operating system - programs are written into punch cards and put into system and system executes them sequentially. Time-sharing - CPU switches between processes frequently, so that multiple users at various terminals can use computer at the same time. This is done by multitasking and multiprogramming. Distributed - Data processing shared among computers and processors accordingly. And it enables the system. Network - runs on a server and manages data, users, security that are connected to it

33. **Explain System V IPC constructs available for Inter Process Communication**	there are 3 constructs available for IPC: semaphores -> for synchronization purpose, message queue -> to send and recieve signals between processes, shared memory -> address space which can be shared and used between them
34. **Explain the following synchronization mechanisms:**	i) Kernel preemption disabling when a process executes in Kernel Mode, it cannot be arbitrarily suspended and substituted with another process ii) Interrupt disabling disabling all hardware interrupts before entering a critical region and reenabling them right after leaving it iii) Semaphores A semaphore is simply a counter associated with a data structure; the semaphore is checked by all kernel threads before they try to access the data structure iv) Spin locks A spin lock is very similar to a semaphore, but it has no process list: when a process finds the lock closed by another process, it "spins" around repeatedly, executing a tight instruction loop until the lock becomes open
35. **Explain the Synchronization problem in Inter Process communication.**	if we say that some global variable V contains number of available items of some system resouce and process A reads V and determines that there is one item avaliable, at this point process B is activated and reads V and decrements it by 1. Then A resumes its execution, because it has already read value of V, decrements the V, and value of V becomes -1. two processes led shared data to an inconsistency state.
36. **Explain what is meant by Device file. State the purpose of device file.**	user visible portion of the device driver interface
37. **Explain what is meant by File Descriptor (Inode). What information does it contain.**	File Descriptor keep all the information about file. File name, owner of the file, size of the file, privileges of the file, type of file and others
38. **Explain what is meant by Hard link. Indicate the command to create a hard link. State the two major limitations of Hard Links.**	file name included in directory is called hard link and command for creating hard link is ln p1 p2 first is hard links can be created among files which are in one filesystem second not possible to create hard link for directory
39. **Explain what is meant by interleaving of Kernel Control paths.**	When kernel control path verifies that request can not be satisfied immediately, CPU interleaves kernel control path
40. **Explain what is meant by Memory Fragmentation.**	When physical memory have enough address space to load new process or file from disk, but the available address space is not located contigiously in physical memory, so new resource can not be allocated from memory even if it is available
41. **Explain what is meant by Process Descriptor.**	Process Descriptor is Process Control Block that keeps all information about current state of the process, when the execution of process finishes, kernel stores contents of several processer registers in the process descriptor
42. **Explain what is meant by Soft link. Indicate the command to create a Soft link.**	Symbolic links or soft links are short files that contain arbitrary path of another file ln -s p1 p2
43. **Explain what is meant by System Call.**	Explit call made to get kernel service via interrupt

44. **Explain what is meant by Virtual Memory. List the advantages of Virtual Memory.**	several processes can be run concurrently, possible to run application whose memory needs exceeds available physical memory, process can execute program whose code is partially loaded, process can share single memory image of library or program, programs are relocatable, programmers can write machine independent code
45. **Fill form BASH**	Bourne Again Shell
46. **First version of Unix was created in ... in 1969**	Bell Labs
47. **Full form of IPC.**	Interprocess Communication
48. **Full form of Unix**	Unics (uniplexed information and computing service)
49. **.......... Generates an exact copy of the current process that differs from the parent process only in its PID.**	Fork
50. **Give a command in linux to reverse a string "WINE Is Not windows Emulator"**	echo "WINE Is Not windows Emulator" | rev
51. **Give a command to delete first line from your file txtfile**	sed -i '1 d' txtfile
52. **Give a command to delete last line from your file txtfile**	sed -i '$ d' txtfile
53. **Give a command to delete lines from 1 to 10 in file txtfile**	sed -i '1,10 d' txtfile
54. **Give a command to delete lines from 20 to the end of the file txtfile**	sed -i '20,$ d' txtfile
55. **Give a command to execute a process in background and indicate a command to move a process from background to foreground.**	put & at the end of the process name and press enter. Then enter fg command to move process from background to foreground
56. **Give a command to get 5 th word in a line in file cfile**	sed -n '1 p' cfile | cut --delimiter=' ' --fields=5
57. **Give a command to get first word in a line in file cfile**	sed -n '1 p' cfile | cut --delimiter=' ' --fields=1
58. **Give a command to get last word from a line in file cfile**	sed -n '1 p' cfile | rev | cut --delimiter=' ' --fields=1 | rev
59. **Give a linux command to assign execute rights to all for the file sfile**	chmod +x sfile
60. **Give a linux command to display line numbers along with each line of text in a file**	cat -n heyy
61. **Give a linux command to display processes sorted on CPU usage**	ps -eo %cpu,fname,pid,ppid,cmd,tty,time | sort -r
62. **Give a linux command to know all the earlier commands entered by you at the terminal.**	history
63. **Give a linux command to make the file fempty only readable for the user and no access rights for group and owner**	chmod o+r,-u,-g fempty

64. **Give a linux command to reverse all lines in your file txtfile**	rev txtfile
65. **Give two commands in linux to display first line of a file.**	sed -n '1 p' fileName or head -1 fileName
66. **Give two commands in linux to display last line of a file.**	tail -1 fileName or sed -n '$ p' fileName
67. **GRUB**	GRand Unified Bootloader and is more advanced than LILO as it recognizes several disk-based filesystems and is capable of reading portions of boot program from files
68. **How can a parent inquire about termination of its children.**	wait() system call allows process to wait until one of its children terminates, it returns pid of terminated child
69. **How do you check how much space left in current drive ?**	df -h .
70. **How do you find for how long your System is up?**	uptime -p
71. **How do you find how many CPUs are there in your system and their details?**	grep "cpu cores" /proc/cpuinfo
72. **How do you find the length of 10 th line in your file txtfile**	sed -n '10 p' txtfile | wc -c
73. **How do you find whether your system is 32 bit or 64 bit ?**	uname -p
74. **How do you find which processes are using a particular file?**	ps -eo fname,pid,ppid,cmd,tty,time
75. **How do you find which process is taking how much CPU?**	top
76. **How do you find zombie process in linux?**	ps -eo state,uid,pid,tty,cmd | grep "^Z"
77. **How do you get (display) last 5 lines from file sfile**	tail -5 sfile
78. **How do you get (display) only first 10 lines from file sfile**	sed -n '1,10 p' sfile
79. **How do you get help about the command "cp"?**	man cp
80. **How do you know if a remote host is alive or not?**	ping google.com
81. **How many privilege levels are in Linux**	2
82. **How will you find which operating system your system is running on in linux?**	uname -v
83. **How will you run a process in background? How will you bring that into foreground and how will you kill that process?**	to run in background -> gedit& , to bring to foregrounf fg
84. **kernel**	kernel is core component of OS, responsible for all major activities of OS, consists of various modules and directly interact with hardware
85. **Linux advantages over commercial competitors**	Linux is free, fully customizable in all its components, runs on low-end, cheap hardware platform, powerful, high standard for source code quality, kernel can be very small and compact, highly compatible with many common operating systems, well supported

86. **Linux file system**	Linux file system is upside down tree structure, root directory denoted by / , files are deferred as leafs and directories as nodes, every directory has root, except parent directory.
87. **Linux's KMA uses .... algorithm on top of a ....**	slab allocator, buddy system
88. **Linux System Architecture**	hardware layer - where all peripheral devices are connected; kernel layer - core component of OS, interacts directly with hardware and provides low level service to high level layer; shell layer - interface to kernel, hiding kernel's functions from users; utilities - utility programs that provide the user with most of the functionalities of OS
89. **List important features of Linux Operating System**	portable, open source, multi-user, multiprogramming, hierarchical file system, shell, security,
90. **List the features a Kernel Memory Allocator should possess.**	It must be fast; minimize amount of wasted memory; reduce memory fragmentation; should cooperate with other memory management subsystem
91. **List the five default actions kernel can perform depending on the type of signal sent to a process`**	terminate the process; write execution context and contents of address space in a file, and terminate; ignore the signal; suspend the process; resume the process execution, if it was stopped
92. **List the types of operating Systems**	1. Batch operating system; 2. Time sharing; 3. Distributed; 4. Network; 5. Real time
93. **........ Loads a new program into an existing content and then executes it.**	Exec
94. **Most filesystems place a limit on the length of a filename, typically no more than .... characters. a) 8 b) 16 c) 64 d) 255**	255
95. **Name the bootloader used by Linux**	LILO or GRUB
96. **OS functions**	Memory management- keep track of memory addresses, allocate and deallocate memory address for process or file from disk when needed Processor management - keep track of processor or processors and allocating and deallocating processes to it when it is needed, set fixed time to process and program called for this job is traffic controller Device management - keep track of all devices connected to kernel, and a program called for this job is I/O contoller. Security - by means of password and other techniques prevent unauthorized access to the programs and data Control over system Performance - recording delays between request for a service and respond from the system Job accounting - keeping time and resources used by various jobs and users Error detecting - production of dumps, traces, error messages, and other debugging and error detecting aids Coordination between user and hardware - coordination and assignment of compiler, interpreters, assemblers and other software to the various users of computer system

97.	**OS properties**	Batch processing - jobs submitted to kernel and done sequentially; multi-tasking - Multiple users use the system simultaneously; multi-programming - multiple programs in memory; real time system - dedicated, embedded system and it must guarantee response to the request in fixed period of time; distributed environment - multiple independent CPUs or processors in computer system where computation logics are distributed among them; spooling - simultaneous peripheral operations on line.
98.	**OS services**	1. Program execution; 2. I/O operations; 3. File System Manipulation; 4. Communication; 5. Error detection; 6. Resource allocation; 7. Protection
99.	**Process**	Process is a program in execution
100.	**Process Control Block**	Process Control Block - keeps all information needed to keep track of process; It keeps process ID, Parent process ID, privilige level of process, state of process, priority, user name, name of executable
101.	**Process Life Cycle**	Start → Ready → Running → Waiting → Terminated
102.	**The process of starting up a computer is known as**	Boot Strapping
103.	**setup() function**	setup() function checkes whether the kernel image is loaded low in RAM (at 0x00010000 address) then moves it to address 0x00001000. Conversly if kernel image is loaded in high RAM, then it does not move. Sets up provisional GDT and IDT. Switches cpu from real mode to protected mode. Jumps to first startup_32() to execute it
104.	**Shell**	Shell program that provides text-only interface for linux and other unix-like OS. It is interface between user and system and it also hides details of kernel operations
105.	**State the four synchronization mechanisms (solutions to synchronization problems).**	nonpreemptive kernel, interrupt disabling, semaphores and spin locks
106.	**State the need for Synchronization**	if kernel control path is suspended and another kernel control path started execution, then second control path must not have access to data structures of first control path unless first kernel control path has been reset to consistent state. otherwise interaction of two control path could corrupt the stored information
107.	**State the purpose of Sync() system call.**	sync() system call, stores all the dirty buffers(whose contents differ from corresponding the disk blocks) in the memory to the disk
108.	**State the purpose of wait() and waitpid() system calls.**	Purpose of wait(), parent waits until one of its child terminates, and when terminates wait() returns pid of that child. waitpid allows process to wait for a specific child process.

109. **State whether the following statement is TRUE or FALSE - In UNIX/Linux, all I/O devices, such as networks, disks, and terminals, are modeled as files, and all input and output is performed by reading and writing the appropriate files. ....**	TRUE
110. **State whether the following statement is TRUE or FALSE - When a process terminates, the kernel changes the appropriate process descriptor pointers of all the existing children of the terminated process to make them become children of init.**	TRUE
111. **The system calls in UNIX is written using which language**	C Language
112. **System call to create a new process is ....**	fork()
113. **System call to load a new program is ....**	exec()
114. **System call to terminate a process is ....**	exit()
115. **There is a file cfile which contains words the, Give a linux command to replace all the occurrences of the in the file with THE?**	sed -i "s/the/THE/g" cfile
116. **There is a file somewhere in your system which has the name cfile. Give a linux command to find that file.**	find / -type f -name "cfile" -print -quit
117. **To avoid the race condition, the number of processes that may be simultaneously inside their critical section is**	1
118. **Unix is which kind of Operating System?**	Multi User, Multi Processes, Multi Tasking
119. **What are the different ways in which a process may react to a signal.**	ignore the signal or asynchoronously execute specified procedure(signal handler). If the process does not specify one of these alternatives, the kernel performs a default action that depends on the signal number
120. **What are the operations performed by second startup() function.**	second startup_32() (process 0 or swapper)function initialize gdt, idt tables and values of gdtr and idtr, initialize segment registers with their final values, fills bss segment of the kernel with zeros. set up provisionals page tables, put global page directory address into cr3 register and enables paging by setting PG in the cr0 register. invokes start_kernel() (process 1 - init) which completes initialization of Linux
121. **What are the operations performed by start_kernel() function**	initialize scheduler, memory zones, final initialization of IDT, softirq_init(), system date time, slab allocator, speed of cpu is determined. kernel thread is executed.
122. **What are the two mechanisms for new process creation?**	fork(); exec()
123. **What are Zombie processes?**	When child process terminates withour acknowledging the parent, then child becomes zombie
124. **What does rm -r * do?**	delete all files and directories recursively in current directory
125. **What happens when parent process terminates, but its children continue their execution.**	Children processes that are continuing their execution become orphan processes
126. **What is an orphan process?**	When parent process terminates before child process terminated

127. **What is an OS? What are the important functions performed by an Operating System**	Operating System is interface between user and hardware. Its important functions are memory management, processor management device management, file management security, control over system performance, job accounting, error detecting, coordination between user and hardware
128. **What is a shell script?**	sequence of instructions executed in order through bash
129. **What is meant by critical region**	Any section of code that should be finished by each process that begins it before another process can enter it called a critical region
130. **What is process group. Give an example.**	process group is a job in which several processes are executed in order and their outputs are shared with each other ls | sort | more
131. **What is shebang construct. Why it is required?**	#!/bin/bash; to alert sytem that a shell script is being started
132. **What is the command to count only the number of characters in the file txtfile**	wc -c txtfile
133. **What is the command to count only the number of lines in the file txtfile**	wc -l txtfile
134. **What is the command to count only the number of words in the file txtfile**	wc -w txtfile
135. **What is the command to find currently running process in your system?**	ps -eo stat,%cpu,fname,pid,ppid,cmd,tty,time | grep -e '^R'
136. **What is the command to find hidden files in your current directory?**	find . -type f -name '.*'
137. **What is the command to find remaining disk space on your system?**	df -h /
138. **What is the default shell when you login to linux?**	/bin/bash
139. **What is the use of the tee command? Explain with an example**	it reads from standard input and write it to standart output or to file: ls -la | tee txtfile
140. **What Linux operating system command would you use to display the shell's environment variables?**	env
141. **Which among the following interacts directly with system hardware?**	Kernel
142. **Which command is used to display the unix version**	uname -r
143. **Which function decompresses the kernel image and loads it into memory starting at physical address 0x00100000.**	first startup_32() function

144. **Which function Initializes the provisional kernel Page Tables contained in swapper_pg_dir and pg0 to identically map the linear addresses to the same physical addresses,**	second startup_32() function
145. **Which function sets up the execution environment for the Linux process (process 0)**	second startup_32() function
146. **Which function switches the CPU from Real Mode to Protected Mode by setting the PE bit in the cr0 control register.**	setup() function
147. **Which is the core of the operating system?**	Kernel
148. **Which of the following functions is (are) performed by the loader**	a) allocate space in memory for the programs and resolve symbolic references between object decks b) adjust all address dependent locations, such as address constants, to correspond to the allocated space. c) physically place the machine instructions and data into memory. d) All of the above
149. **Which of the following statement is FALSE ?**	Shell takes care of inter process communication
150. **Which process is created by start_kernel() function.**	process 1;
151. **Which process loads the gdtr and idtr registers with the addresses of the GDT and IDT tables**	process 0;
152. **Which routine copies the Kernel image into memory.**	Boot loader copies kernel image into memory
153. **Which routine loads the Boot Loader into memory.**	BIOS procedure loads boot loader into memory
154. **Write a linux command to create an empty file fempty in your directory (without using any editor)**	touch fempty
155. **Write a linux command to list all the symbolic links in the current working directory**	find . -type l
156. **You have txtfile file created in your directory. Give a linux command to display second column values from the fiile.**	cut --delimiter=' ' --fields=2 txtfile
157. **Your application home directory is full? How will you find which directory is taking how much space?**	du -h