LINUX QUESTIONS

1. About IPC mechanisms, difference between message queues and shared memory?
2. What is RTOS, its featuers, interrupt latency, what is interrupts?
3. what is device driver? Why we need it? Do we need device driver to access memory?
4. what is scheduling algorithm ? what scheduling algorithm unix uses? RTOS uses?
5. explain scheduling algorithm?
6. How is RTOS time deterministic?
7. What threads will share?
8. what fork returns to child - Ans 0
9. A question on Raw socket
10. Which IPC we can use between 2 PC's?

a. Semaphores b. Pipes c. Sockets d. None

1. What is software Interrupt and Hardware interrupt
2. About watch dog timer, How can we convert the the Watchdog timer interrupt(Its a software interrupt)to hardware interrup
3. signals are part of IPC yes/no. Justify.
4. what is multithreading and multiprogrmming.
5. Explain concept of multtreading using CPU.
6. write a program to print the link data from the last node.
7. what is memory fragmentation.
8. what is static and their scope as variable and function.
9. where static, auto and globals are stored.
10. what is process and thread.
11. what is pipeline.
12. write a program to swap two numbers.
13. Explain fork.
14. Explain diffrent IPC mechanisms.
15. Difference between process and thread.
16. some questions on threads and signals.
17. Questions on Mutex and semaphore
18. What is the difference between MUTEX and SEMAPHORE.
19. Write a complete program using pipes to communicate between two processes?
20. What is RTOS? Why is it called real-time? Why is it efficient than other OSes?
21. What are different IPC mechanisms? Which ipc do u prefer and why? What is deadlock? Which ipc did u use in project and why? How is synchronization achieved, give example? How does Mutex work? Did you work on shared memory? What is critical section? How do u avoid deadlock?
22. You have data coming from bus and Program Status Register will indicate whether data is available and you are processing 500 bytes of data and when you are processing 250 th byte,if PSR will indicate data is available, how do u handle processing data(either in drivers or normal way) with PSR indication?
23. Architecture of the Linux
24. Write a program to implement semaphore operation.
25. Daemon, orphan, and zombie process.
26. Process data structures( process table, pregion table, region table ).
27. How do u broadcast using a system call
28. Explain Dup system call and how u will redirect the output to a file instead of std output fork(), vfork() system calls( differences, when to use vfork), Copy On Write( COW). mount, open, exec, wait, signal system calls.
29. Signals, signal handling, how to send signals
30. ( kill command or system call).
31. Shared memory( with system calls: shmget, shmat, etc )
32. Semaphores( used for synchronization, like access to shared memory)
33. What is advisory lock.
34. What is mandatory lock
35. Who will take care about mandatory locking ?
36. What is spin lock ?
37. What is dead lock ?
38. In which case we have to use spin locks
39. Is it possible to have dead lock situation where there is only one process and only one resource What is paging ?
40. In case of paging system suppose reading is happened at the end of one page ,How it locates the next address to read ?
41. If one code is written to the system which does not support paging, What changes has to be made to run it on the system which supports paging ?
42. How rm works ?
43. What is the difference between VxWorks kernel and Linux kernel ?
44. Why DOS is not multitasking system and Linux is multitasking ?
45. Multitasking is implemented in software level or hardware support is also needed? How open system call works in kernel level ?
46. Where monolithic kernels are used ? Where micro kernels are used ?
47. Write a program to generate 5 childs they run parrellaly?
48. What are the different states of the process?
49. What is the difference b/w zombie and orphan?
50. What grep command will do?
51. Write the kernel architecture?
52. What is the different types of RTOS?
53. How can you load the device driver modules?
54. Did you compile a kernel?
55. What are the different driver modules?
56. Which commmand you used in linux to load the modules?
57. Explain the shared memory?
58. Did you write any kernel programs?
59. What is the difference b/w hardware and software interrupts?
60. What is privilaged mode ? Give examples?
61. Explain the interrupt handling?
62. How the system call execution takesplace?
63. Explain the context layers of the process?
64. What is Demand paging ?
65. Did you read any linux kernel book?
66. What are the File systems of LINUX?
67. What is the difference b/w EXT2 and EXT3?
68. How are u going to check the default shell in your system ?
69. How can u change the default shell ?
70. What does this first line #!/bin/bash in the shell script indicate ?
71. Which shell will interpret if u use #!/bin/csh as the first line in the
72. shell script ? and which will interpret if u run script without using this
73. first line ?
74. Explain the Process Data Structure. What happens when a process is created.
75. What are signals? How are they implemented ?
76. How are default signals handled.?
77. What are different modes of signals?
78. What is the difference between dup and dup2?
79. How to substitute a string in shell Programming ?
80. In shell , how will you find the details of devices which have been mounted ?
81. Write a program to generate 5 childs they run parallel?
82. About umask
83. What is spin lock ?
84. How open system call works in kernel level ?
85. What are the different states of the process?
86. What is the difference b/w zombie and orphan?
87. Write the kernel architecture?
88. Did you compile a kernel?
89. What are the different driver modules?
90. Which commmand you used in linux to load the modules?
91. Explain the shared memory?
92. Did you write any kernel programs?
93. What is the difference b/w hardware and software interrupts?
94. What is privilaged mode ? Give examples?
95. Explain the interrupt handling?
96. How the system call execution takesplace?
97. Explain the context layers of the process?
98. Which command you used to check shared library?
99. Explain virtual address mapping?
100. Did you read any linux kernel book?
101. What are the File systems of LINUX?
102. What is the difference b/w EXT2 and EXT3?
103. What is journaling?
104. What fsck will do? --> check and repair filesystems
105. Explain the imlplementation of Linux file system?
106. What are the contents of inode?
107. What is Copy on write? Where it is used?
108. What is delayed write?
109. What is PCI and how many devices it supports?
110. What is the scheduling algorithm used in unix?
111. Explain physical memory layout. Explain different segments.
112. Where are the shared libraries loaded?
113. When are the shared libraries loaded in main memory?
114. If two processes uses one shared library, how the other process come to know that particular library is already loaded in memory?
115. What are signals? how they have been implemented?
116. When is a signal handled?
117. What happens if same signal is sent more then once?
118. Explain wait system call.(arguments,return values)
119. How does the parent process comes to know the child exited.
120. (process table entry)
121. How are default signals handled?
122. What are different modes of signals?
123. Can we have lseek on a tape device?
124. How system call is invoked?
125. When SIGBUS signal is generated?
126. What is TLB?
127. What is shared memory?
128. How to synchronize processes when they access shared memory?
129. What are the types of semaphores?
130. How will you synchronize n processes from accessing r resources?
131. Virtual Memory
132. Problem with parallel execution

a=b+c

d=x+y

s=a+d

how will u make them to execute in Parallel

ans:

P1: a=b+c

P2: d=x+y

Then any of them to do s=a+d

1. What do u know about multi-threaded applications?
2. In which case we have to use spin locks
3. In case of paging system suppose reading is happened at the end of one page. How it locates the next address to read ?
4. If one code is written to the system which does not support paging, What changes has to be made to run it on the system which supports paging ?
5. Any real life use where multiple resources are accessed in synchronization
6. How and Where counting semaphores are used explain?
7. What is the difference between VxWorks kernel and Linux kernel ?
8. Why DOS is not multitasking system and Linux is multitasking ?
9. Multitasking is implemented in software level or hardware support is also needed?
10. Where monolithic kernels are used ? Where micro kernels are used ?
11. What is the different types of RTOS?
12. How do you convert Linux as a RTOS?
13. What is virtual address and physical address?
14. How virtual address is mapped into physical address? Which hardware does it?
15. How signal handling is done
16. How interrupts are handled
17. what to do if I want to delete the process after executing.
18. Can fork be done in threads and if done how and what will happen.
19. Difference between threads and process.
20. Different IPC mechanisms.
21. what are the unix concepts u know?
22. list the ipc's u know?
23. diff bw pipe and sockets?
24. (before getting into this thread concepts he asked me something but i dont remember .after that he asked..)
25. how can i create a thread?
26. What is context switching?
27. What is preemption?
28. diff bw unix and windows.(some cross q's)
29. diff bw unix and linux.
30. what do u know abt unix internals?
31. What is the difference between Binary and Mutual Exclusive Semaphore?
32. What is Priority Inheritance?
33. all ipc explanation with system calls (pipes, msg q, shared mem, semapore, socket)
34. how open sytem call work( with file table, UFDT, IT)
35. how read write system call work.
36. Filesystem structure, Kernel architecture
37. signal handler in the case of child ( when the signal wll be handled ) if parent recieve signal and then creted child and child executed first then when signal will be handled and who wll handle ?
38. int main ()

{

fork ();

fork ();

fork ();

printf (" hi\n");

}

how many time it wil print hi and why ?

Ans: power(2,n), n is number of forks;