

OS Midterm Questions

1) Suppose you have opened a file for reading that has a lot of text. Now you call the system call `read()` to read its content, with a buffer size of 100. What will the `read()` function return?

- (A) 0
- (B) 100
- (C) String of length 100
- (D) Anything between 0 and 100

Option: _____D_____

2) *The function of the accumulator register in Intel X86 is as follows:*

- (A) Performing ALU operations and storing the results.
- (B) Performing memory read/write operations.
- (C). Deciding if the outcomes of ALU operations for the sake of making appropriate jumps and / or returns.
- (D). None of the above.

Option: _____a_____

Resolution: Either A or D are fine.

3) Suppose you call the `write()` system call to write "ABCD" to an empty file and then immediately a different program reads the same file. What is the expected behavior?

- (A) It will read "ABCD"
- (B) It will find the file empty
- (C) It will get an error
- (D) Either (A) or (B)

Option _____d_____

4) Suppose you open a file. You then call the `fork` system call. Can you write to the same file from the child process using the same file descriptor?

- (A) Yes
- (B) No

Option _____a_____

5) In Q3, what if you call the `exec` system call?

(A) Yes

(B) No

Option b

6) In Q3, what if you call the clone system call?

(A) Yes

(B) No

Option a

7) Suppose a parent process quits without waiting for child process. Who inherits the child process?

(A) The init process

(B) The closest sub-reaper process on the process tree

(C) The kthreadd process

(D) None of the above

Option b

Resolution: Either A or B

8) How does our system deal with paging and segmentation where it is in real-mode?

(A) Using a smaller page table and segmentation register

(B) Using a smaller page table, but without using segmentation register

(C) Does not use page table, but uses segmentation register

(D) None of the above

Option d

9) What is the output of the following program?

```
void swap (int a, int b) {  
    int c;  
    c = b;  
    b = a;  
    a = c;  
}  
  
int main (void) {  
    int x=3, y=4;  
    swap (x, y);  
    print ( "%d %d" , x, y);  
}
```

- (A) 4 3
- (B) 3 4
- (C) 3 3
- (D) 4 4

Option _____ b _____

10) What is the output of the following program?

```
void swap (int a[]) {  
    int c;  
    c = a[1];  
    a[1] = a[0];  
    a[0] = c;  
}  
  
int main (void) {  
    int x[]={3, 4};  
    swap (x);  
    print ( "%d %d" , x[0], x[1]);  
}
```

- (A) 4 3
- (B) 3 4
- (C) 3 3
- (D) 4 4

Option _____ a _____

11) What is the time complexity of current Linux scheduler?

- (A) $O(1)$
- (B) $O(\log n)$
- (C) $O(n)$
- (D) $O(n^2)$

Option _____ b _____

12) Which of the following problems does round robin scheduling suffer from?

- (A) starvation
- (B) lack of adaptation to workload
- (C) priority inversion
- (D) none of the above

Option _____ b _____

13) Suppose you create a thread using the pthread interface. Which of the following system

calls is used by Linux to implement this action?

- (A) fork
- (B) clone
- (C) exec
- (D) none of the above, as pthread is a library function

Option b

14) Suppose you try to boot your computer without a RAM? At what stage will this problem be flagged?

- (A) real-mode operation
- (B) extended-mode operation
- (C) Loading of GRUB
- (D) POST

Option d

15) Suppose we write a program hello.c in C on our personal computer. We first (1) run the preprocessor to get hello_ex.c, (2) then compile it to assembly get hello_ex.S, (3) assemble it to get hello_ex.o and (4) finally link it to get hello.out as executable. Suppose you move all these files starting from hello.c to a Raspberry Pi, with ARM processor. From which step would you need to re-run to ensure that the executable runs on the Raspberry Pi?

- (A) Step (1)
- (B) Step (2)
- (C) Step (3)
- (D) You cannot run the same program on Raspberry Pi

Option b

16) Suppose you want to write exactly 100 bytes to disk using write system call. Which of the following calls to write would lead to the fastest write?

- (A) Buffer size of 1, with O_SYNC flag
- (B) Buffer size of 100, with no additional flags
- (C) Buffer size of 1, with no additional flags
- (D) Buffer size of 50, with O_SYNC flag

Option b

17) *Suppose a child process does not want the parent to wait for its completion. Is this possible, and if so, how?*

- (A) The child process can send a signal to the parent process
- (B) The child process can call another fork and then quit
- (C) The child process can call wait

(D) This is not possible using any technique

Option_____d_____

18) Suppose you have a global variable x initialized to 0. You create a total of three threads, each of which increment the value of x by 1. What will be the value of x at the end of the program?

(A) 3

(B) 2

(C) 1

(D) Any one among 1, 2 or 3.

Option_____d_____

Resolution: Remains (D)

19) Which of the following assembly language instructions are incorrect?

(A) ADD rax, [rbx]

(B) ADD [rax], rbx

(C) ADD [rax], [rbx]

(D) None of the above

Option_____c_____

20) Which of the following tasks is not supported by a general purpose computer?

(A) Soft real-time tasks

(B) Interactive tasks

(C) Hard real-time tasks

(D) All of the above are supported

Option_____c_____

21) Suppose you initialize a pointer with 0, and then try to access the memory space stored in the pointer in the next instruction. What will you get on executing the program, and why does it happen?

(A) An asynchronous signal SIGINT

(B) Segmentation fault, caused by an interrupt

(C) Returns data used by kernel

(D) None of the above

Option _____ b _____

22) *Fork(), pthread_create() and vfork() differ in the following ways:*

- a. Vfork() does a lazy copy while fork() creates copies of all program memories – code, stack, data, BSS, heap and RODATA.
- b. pthread_create() and vfork() do a lazy copy while fork() creates copies of all program memories – code, stack, data, BSS, heap and RODATA.
- c. Both vfork() and fork() do the same thing. Pthread_create() however is different, in that it is used for thread creation.
- d. None of the above.

Option _____ d _____

23. *The key purpose of the timer interrupt is:*

- a. For maintaining the current clock time (based on users' timezone).
- b. For invoking the task scheduler to switch process context.
- c. For setting up alarms and event timeouts.
- d. None of the above.

Resolution: Option _____ b _____

24. Which of the following is true about the 'init' process.

- a. It is a kernel thread.
- b. It is a background process (zombie) and no PCB is allocated for it.

- c. It does not have any PID, as it is a kernel thread, which anyways do not have PIDs associated.
- d. It does not itself handle any signals, but allows child processes to do so.

Option_____d_____

25. Which of the following is NOT true about most process scheduling schemes:

- a. May take into consideration potential race conditions.
- b. Takes into consideration blocking system calls.
- c. The task scheduler also needs to take care of process memory allocation, so as to improve task scheduling efficiency, upon process creation.
- d. Usually makes use of process priorities to allocate appropriate timeslices to various processes.
- e. A&B
- f. A&C
- g. B&C
- h. B&D
- i. All the above
- j. None of the above

Option_____f_____

26. Assume a functional linux system (kernel > 2.6.23), with the following processes:
- Processes A,B and C, each with nice value of 0 each.
 - Processes D and E with nice values of -5 and -7 respectively.

Assume that the value of weight is computed using the formula $W = 1024/(1.25)^{(\text{nice})}$.

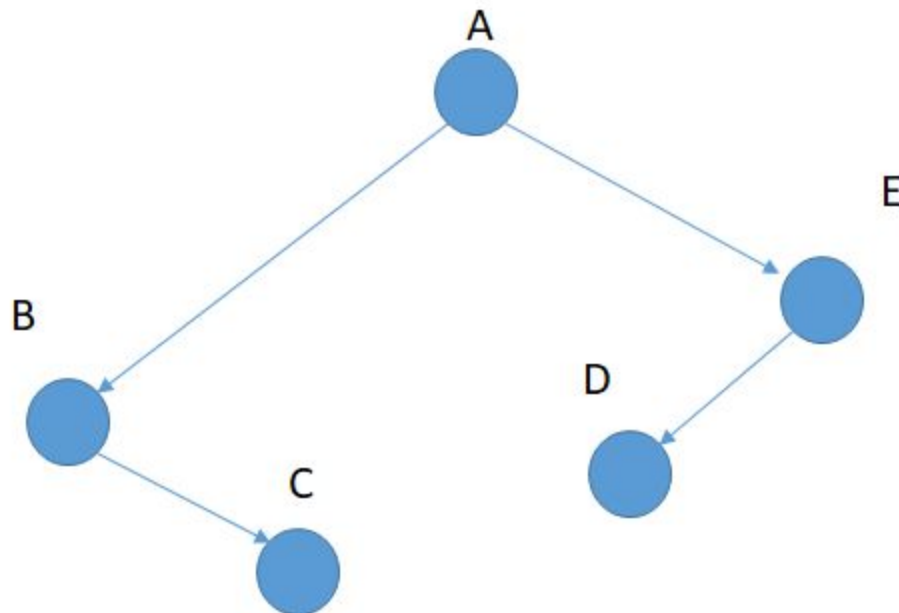
Also, assume that the *target latency* is selected to be 100 usec.

You need to do the following:

- Compute the *vruntime* values for the current state of the processes.
- Draw the sched_entity RB tree for the above processes (with their current *vruntimes*).

Soln:

The weights of each of the processes are 1024,1024,1024,3125, and 4882.813 respectively. The CPU percentage for each of the processes are 9.24%,9.24%,9.24%,28.2% and 44.06% respectively. Thus the vruntimes for the processes are 9.24us,9.24us,9.24us, 28.20us and 44.06us respectively.



Explain the rationale behind computing the vruntimes for each case and how you drew

the RB tree for the same.

27. Socketpair() differs from pipes() in the following way:

a. Socketpair is like a ``easier to use'' version of pipes whereby the processes need not take care of manually managing various descriptors.

b. Pipes() are bi-directional while socketpair provides uni-directional communication.

c. Socketpair can be used for byte oriented, fixed size messages while pipes can be used for arbitrary length messages.

d. Socketpairs are associated with FIFO file, while pipes aren't

Option_____a_____

Resolution: Everybody gets marks

28. FIFOs differ from sockets in the following ways:

- a. FIFOs work through message passing schemes while sockets communicate via byte streams.
- b. Read and write to a FIFO cannot be pre-empted as it involves pre-empting and context switching two processes, unlike the case of sockets where the messages are queued.
- c. Bytes are queued in a FIFO, unlike that in a socket, where unread messages are lost.
- d. Because of (c) FIFOs cannot be used for real time communication, while sockets can.
- e. None of the above

Option_____e_____

29. FIFOs differ from shared memory in the following ways:

- a. They are implemented in the same way, though only appearing semantically differ from the application programmers point of view.
- b. Unlike FIFOs, shared memories are like pipes and are shared between parent and child processes only.
- c. FIFOs cannot be used with blocking system calls, while shared memory can be.
- d. None of the above

Option_____d_____

30. Which of the following cannot be used while modifying or adding code to the Linux kernel?

- a. C data types
- b. structure
- c. Library function
- d. Macros

Option_____c_____

31. Which of the following can run concurrently but not in parallel in Linux?

- a. Child and parent processes
- b. Threads
- c. User-level threads, created without invoking system calls
- d. Distinct processes

Option_____c_____

32. What of the following is one of the disadvantages of having a monolithic kernel, like Linux?

- a. Slow performance
- b. Requires assembly programming
- c. Lack of support for isolation of execution among modules
- d. Unsuitable for embedded systems

Option_____c_____

33. Suppose you are an ordinary user with no root privileges. You decide to create a process. What is the range of nice values you can change the process to?

- a. -20 to 19
- b. 0 to 19
- c. 10 to 19
- d. -20 to -10

Option_____b_____

34. Suppose a new Intel processor adds an additional opcode which the Linux kernel wants to use. How would it be able to use it?

- a. Using a C statement
- b. Using a system call
- c. Using inline assembly
- d. Cannot use it until compiler supports

Option_____d_____

35. Suppose you write a new device driver for the Linux kernel. How would you use it to access the hardware?

- a. Compiling the driver module and loading it
- b. Compiling the driver module, but loading it while rebooting
- c. Compiling the driver module, linking it and use the new kernel
- d. Only compiling the driver module

Option_____a_____

36. Suppose you have an array of size 10. You try to access position 15 in C. What behavior do you expect?

- a. Segmentation fault
- b. Return of garbage value
- c. Process would abort
- d. Either Segmentation fault or return of garbage value

Option_____d_____

37. Suppose you allocate memory in a C program using malloc. In which segment of memory does it get allocated?

- a. Stack
- b. Heap
- c. Text
- d. BSS

Option_____b_____

38. Which of the following system calls returns control from the function only in case of failure?

- a. Fork
- b. Clone
- c. Exec
- d. Wait

Option_____c_____

39. Which of the following tools is NOT an essential component while building a user-level C program?

- a. Preprocessor
- b. Compiler
- c. Linker
- d. Makefile

Option d

40. Suppose you want to run a (a) 32-bit executable program on a 64-bit Intel processor, and (b) a 64-bit executable program on a 32-bit Intel processor. Which of the following is correct, without using any simulator or emulator?

- a. Both (a) and (b) are impossible
- b. (a) is possible, but (b) is impossible
- c. (a) is impossible, but (b) is possible
- d. Both (a) and (b) are possible

Option b