

CS330 - Quiz 1

Total: 15 marks

Duration: 20 mins

open books and notes, no mobile phones, friends disconnected during exam

Be precise, no marks for vague answers

Roll No:

Q1. Select the correct answer(s) for the following questions. No partial marks. (2×3)

- (i) You have added code at the beginning of the functions mentioned in the question to print the stack pointer every time the function is executed. Which of the following statement(s) are true regarding the value of stack pointer?
- (a) For a given user-level function, value of stack pointer *can be* different.
 - (b) For a given OS-level function which is exclusively used by a system call handler (no other OS code calls this function), value of stack pointer *can be* different.
 - (c) For two different OS-level functions, the value of stack pointer *can be* same.
 - (d) For a given OS-level function handling a particular interrupt, value of stack pointer *must be* same.
 - (e) For two different functions from two different application, value of stack pointer *can be* same.

Ans: a, b, c, e

- (ii) In a uniprocessor system, in which of the following case(s), a process executing in user mode can cause an entry into the OS?
- (a) accessing a general purpose register like **RAX**
 - (b) executing a **JMP** (jump) instruction
 - (c) decrementing an unsigned integer value stored in a register beyond zero
 - (d) executing a **printf()** statement
 - (e) returning from a function

Ans: b, d, e

- (iii) Which of the following statement(s) are true in a system with only segmentation (no paging)?
- (a) Two segment registers may point to the same segment descriptor
 - (b) If an ISA supports one segment register, run-time binding is not possible
 - (c) If an ISA supports one segment register, intra-application isolation is not possible
 - (d) If an ISA supports one segment register, isolation across applications is possible but OS memory can not be isolated
 - (e) If an ISA supports one segment register, inter-application isolation and OS isolation can be achieved

Ans: a, c, e

Q2. True or False with justification. No marks without proper reasoning. (2 × 3)

- (i) In a machine with a single CPU, an application executing an infinite loop will freeze the system.

Ans: *False*, The OS will get control of CPU resource to handle interrupts, especially the timer interrupts. The interrupt handler can switch out or kill the process.

- (ii) Without compromising isolation, two applications can communicate among themselves without the OS support.

Ans: *False*, when two processes are sharing a part of resource, to achieve control on the extent of sharing and check access to unshared part of the resource, the OS support is required.

True (less practical), in a scenario when the entire resource is only used by the two application and no other application is using the resource.

- (iii) In a hierarchical paging system, for any given virtual address, page table entries at each level must have same access permissions (read, write and execute).

Ans: *False*, page table entry at upper level dictates the permission for a larger VA space which might require different permissions. For example, a level-1 entry in 48-bit address covers 512 GB memory range which can be shared by code and data, and require different permissions.

Q3.

```
void initialize_array(void)
{
    int array[4];
    int ctr=4;
    do{
        ctr--;
        array[ctr] = 0;
    }while(ctr >= 0);
    printf("%d\n", array[3]);
}
```

What will be the output of the above function in X86 systems? Explain. (1×3)

Ans: The program will execute infinitely and will not terminate.

This behavior is primarily due to two reasons,

- The loop will terminate when the value of $ctr = -1$, which implies the program will access $array[-1]$.
- $\&array[-1] = \&ctr$: Since the stack grows from higher address to lower address