Roll. No.: _______________ Name: _________________________________ Section: ____

CSE638: Graduate Systems
Midsem Winter 2025 (SOLUTIONS)

Date: February 23, 2025; Duration: 60 mins (10 am to 11 am)

Total marks: 30

Instructions:

(1) Read the questions carefully and answer.
(2) **Answer to the point as much as possible.** Include the important technical details.
(3) If you make practical & appropriate assumptions, mention them clearly.

Q.1. "When an interrupt occurs, the process switches from user mode to kernel mode for interrupt processing. To start the execution of interrupt code, the system has to fetch the process image of OS into the memory" State True or False; No marks without justification. **[2]**

Ans: False; The memory image of each process includes the OS image (at the high end of the virtual address space)

Q.2. Given that a regular process is currently running on the CPU. State at least FOUR distinct events (i.e., type of events) when the Operating System takes control of the system. **[4]**

Ans:

(1) Hardware Interrupt
(2) Fault occurs during program execution (buffer overflow, divide by zero)
(3) The program makes a System call
(4) Timer interrupt (the scheduler kicks in)

Q.3. For each of the following variables declared in a C program, state which part of the memory image of the process will contain the memory allocated to the variable. **[4]**

(a) A global variable declared outside any function in the program.
(b) A static variable declared inside a function definition.
(c) An integer variable declared inside the main function.
(d) The variable "ptr" in the following line of code located inside the main function
int *ptr = malloc(sizeof(int));

Ans:

(a) Code + compile-time data (Just "compile-time data" is also ok)
(b) Code + compile-time data (Just "compile-time data" is also ok)
(c) Stack
(d) Stack

Q.4. You are using a Virtual Machine (VM) for system virtualization. The application running inside a VM accesses the system resources such as CPU, memory, and I/O. Answer the following questions related to CPU virtualization across VMs. **[2+2+3=7]**

(a) Assume traditional trap-and-emulate mechanism is implemented (without hardware-assisted virtualization). Will the system work correctly under the following scenarios? If "yes" tell how, and if "no" tell why? **[2]**

(i) CPU instruction "X" requires access to system resources.

Roll. No.: _______________ Name: _________________________________ Section: ____

(ii) CPU instruction "Y" can run in both "privileged" and "non-privileged" mode with a different outcome.

(b) Imagine a scenario where your VM always runs applications that execute **unprivileged instructions 99.9%** of the time, and **privileged instructions 0.1%** of the time. Would you prefer "paravirtualization" or "full virtualization" (note: you have no other choice)? Why? **[2]**

(c) Suppose QEMU/KVM architecture is used for hardware-assisted CPU virtualization. What are the major steps involved when a Guest application executes a privileged instruction? Include the data structures used to store context. **[3]**

Ans:

(a) (i) Yes, CPU instruction "X" will run correctly since the VMM will handle the privileged operation.
(ii) No, CPU instruction "Y" is a sensitive instruction that results in different outcome when run in "privileged" vs. "non-privileged" mode.

(b) **"Para virtualization"** for performance, but **"Full virtualization"** for flexibility. [Your answer should make the right argument]
Paravirtualization requires changes to the Guest OS that converts all privileged OS instructions to cause a trap to VMM. It is not flexible but it does not have runtime overheads.
In full virtualization, VMM tests for each CPU instruction for privileged operation at runtime. This costs runtime overhead even if the CPU instruction does not require privileged access.

(c) Guest OS (ring 0) and Guest application (ring 3) runs in CPU's VMX mode. QEMU (ring 3) and kvm (ring 0) runs in CPU's root (i.e., non-VMX) mode. When a guest application executes a privileged instruction, the system mimics the trap-and-emulate design.
(i) Guest application traps into the Guest OS (ring 0, VMX mode)
(ii) Guest OS triggers VM exit (switch from VMX mode to root mode)
(iii) The guest context is saved in VMCS and the host context is restored from VMCS
(iv) Based of the VM exit reason, KVM handles the VM exit and performs resource management tasks OR KVM returns to QEMU.
(v) If the control does not return to QEMU, KVM resumes the Guest VM by restoring the Guest context & switching back to VMX mode. The Guest OS returns the control to the guest application.

Q.5. Consider an architecture that uses **N-level hierarchical paging**, where all page tables (traditional (or regular), extended, shadow, and so on) have **"N" levels**. A hypervisor on this system runs a **single guest VM**, which in turn has **TEN active processes**. **[4]**

(a) If the hypervisor uses **extended page tables (EPT)** for memory virtualization
(i) How many extra page tables does the hypervisor maintain to virtualize the guest, in addition to the page tables maintained by the guest VM itself?
(ii) How many memory accesses (approximately) must the MMU perform in order to "walk the page table" and translate ONE address?

(b) If the hypervisor uses **Shadow page tables** for memory virtualization
(i) Same Question as 5(a)(i)

(ii) Same Question as 5(a)(ii)

Ans:

(a) (i) **ONE** EPT (maps GPA to HPA)
(ii) **N^2** memory accesses; for each guest page table access there is one "page table walk" into the extended page table.

(b) (i) **TEN shadow page tables**: ONE shadow page table per process that maps GVA to HPA
(ii) **N** memory accesses; one for each level of page table

Q.6. Suppose you are running a web application that communicates with the web server over the Internet over a 1 Gbps network cable. If you somehow know the following. **[3]**

(a) The bottleneck bandwidth from your machine to the web server is 10^6 bits/sec
(b) The time required to send your data is 1 millisecond (i.e., ms)
(c) The time requires to receive an ACK is 1 millisecond

What should be the TCP window size for the web application to maximise the network throughput?

Ans: Bandwidth Delay Product (BDP) = Bandwidth * RTT = 10^6 bps * (1+1) ms = **2* 10^3 bits (i.e., 2000 bits or 2 Kbits)**

Q.7. Complete the following code snippet by filling in the blanks: **[6 * 0.5 = 3]**

```
typedef struct {. . .} arg_t; // Dummy; Don't fill this
void *thread_func(void *arg) {
        . . . // Dummy; Don't fill this
}
int main(int argc, char *argv[]) {
    pthread_t p;
    arg_t args = { 10, 20 };

    printf("running 'thread_func' with 'args'\n");
    int rc = pthread_create(&p, _____, _____, _____);// (A)

    printf("waiting for 'thread_func'\n");
    __________(_____, _____); // (B)
    printf("done\n");
    return 0;
}
Ans. (A) ____, _____, _____ (B) ______(______, ______)
```

Ans.
(A) NULL, thread_func, &args
(B) pthread_join(p, NULL);

Q.8. What system calls will be used for the following execution sequence: **[3]**

(a) Process "P1" wants create a new process "P2" and place it into a new namespace "ns1"
(b) Process "Q1" is in the default namespace. Q1 wants to create a new process "Q2" and place it into namespace "ns1"

(c) Process “P2” wants to create a new process “P3” in namespace “ns1”

Ans. (a) clone(); (b) setns(); (c) fork()

_________________________**THE END**________________________