New York Institute of Technology
CSCI 330/509-M02 - Operating System
1st Quiz - Fall 2017
Instructor: Prof. Susan Gass

PART (I) Fill-in-the-blank Questions:

1. **kernel** is the one program running at all times on the computer.
2. **Virtualization** allows operating system to run applications within other operating systems.
3. **Virtual memory** allows execution of processes not completely in memory.
4. In Symmetric Multiprocessing (SMP), processors perform **all** tasks.
5. A **timer** can be used to prevent a user program from never returning control to the operating system.

PART (II) Short-Answer Questions

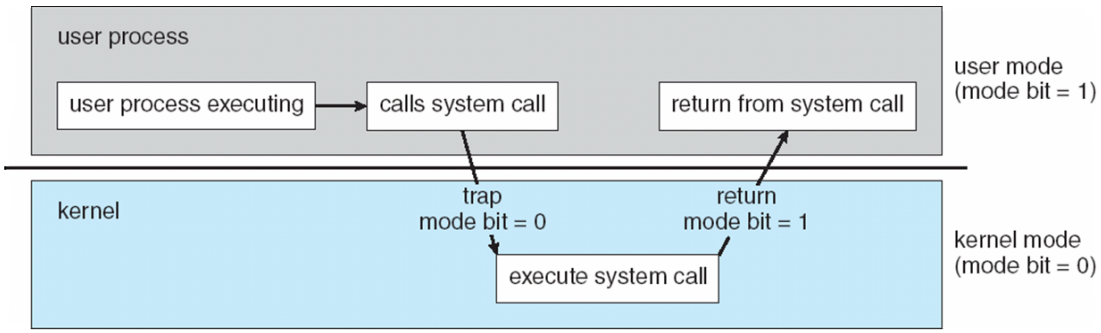
6. Why should devices (with the exception of CPU) have device controllers?

- It's uses as an interface between device and the operating system.
- Device controller is a hardware with a local buffer (which is a special-purpose register), that's used for exchange of data between the device controller's local buffer and memory through the CPU. Communication with CPU is by interrupts, where device controller notifies the CPU that it has finished an operation, by causing an interrupt.
- I/O is from the device to local buffer of device controller

7. List and briefly describe the two modes of operating in a system.

User mode (mode bit = 1): User application/Process is executing. However, since there is no direct access to hardware (including memory), requests for OS services via system calls, causes transition from user mode to Kernel mode.

Kernel mode (mode bit = 0): Handles system calls (which are interfaces to OS services)



8. Briefly describe the purpose of interrupts and their implementation.

An interrupt is a signal to the CPU, generated by hardware or software, notifying the CPU that an event has occurred that needs its attention.

Hardware interrupts: are caused by either internal device such as device controller or external devices such as keyboard, mouse, printer, scanner, sound card, camera, etc., requesting CPU services. For example, device signaling that it has received some data (e.g., keystrokes on the keyboard or data from network card); or it has just completed a task such as transferring data between the hard drive and memory.

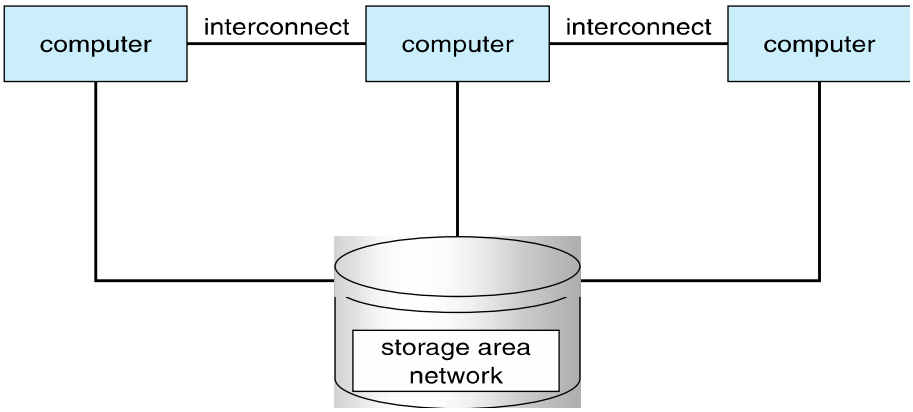
Software Interrupts: are generated by user program to handle a traps or exceptions, which is sent to OS and it's run in privileged mode. Or interrupts sent to and from disk controller to request reading or writing of data to and from the disk. Software interrupts are also causes by programs requesting a service from OS via system calls.

Interrupt Handling: For handling interrupts, CPU suspends its current activity, saves its state, an interrupt vector which contains the addresses of all the Interrupt Service Routines (ISRs) is looked up to locate the appropriate ISR to run. After interrupt is handled, CPU continues its activities from where it was interrupted.

9. Briefly describe clustered systems and why they are considered high-availability systems?

A collection of systems connected together on a network, sharing a storage via (SAN). They provide high-availability service which survives failure due to redundancies in these systems. So, if a node fails other nodes will take over the failed node's task.

Asymmetric clustering: where one machine in hot-standby mode, to look out for failed nodes and handle the task.
Symmetric clustering: where multiple nodes are running and monitoring each other.



Good Luck!

New York Institute of Technology
CSCI 330/509-M02 - Operating System
2nd Quiz - Fall 2017
Instructor: Prof. Susan Gass

PART (I): Fill-in-the-blanks Questions:

1. The Device controller is responsible for moving the data between the peripheral devices that it controls and its local buffer storage.
2. Microkernel provides more security and reliability, since most services are running as user.

3. Modules allow operating system services to be loaded dynamically.

4. A modular kernel may be the best of the current operating system design techniques, because

In addition to modules being loaded dynamically, when needed, adding or modifying the existing modules won't require modifying the kernel. Modules can communicate with each other, which makes it more flexible than the layered approach.

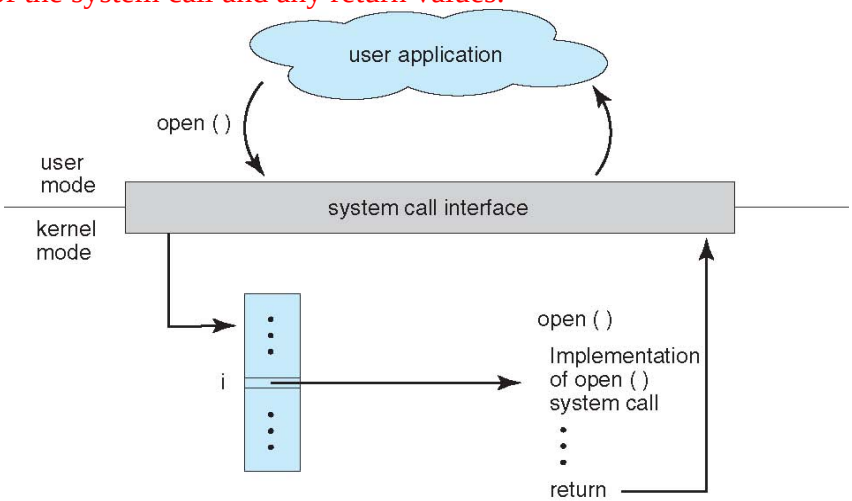
PART (II): Short-Answer Questions

5. Briefly describe the purpose of systems calls and their implementation.

Systems calls are requests from user applications for OS services, such as reading a file, writing to some device, or maybe even running another program.

Implementation:

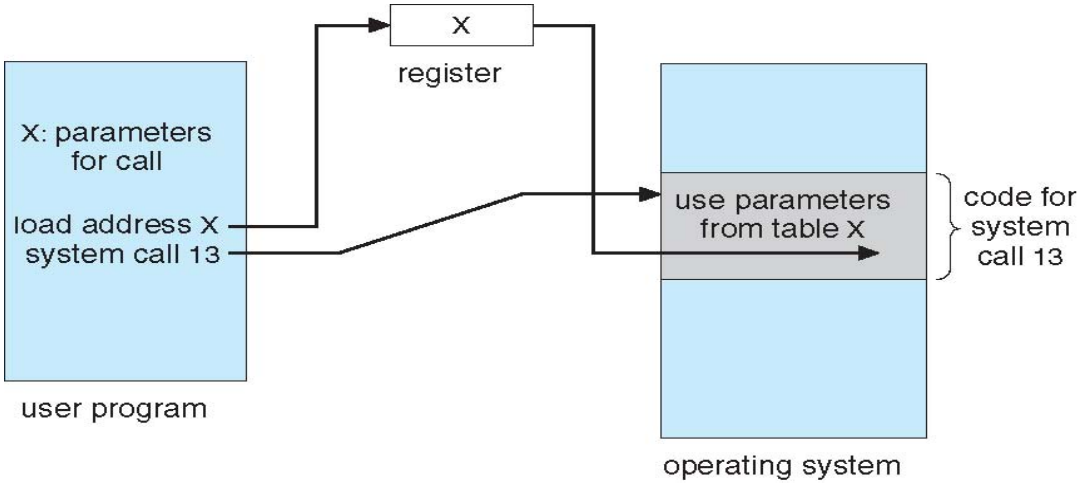
- A number is assigned to each systems call.
- System-call interface maintains a table indexed according to these numbers.
- System-call interface invokes the intended system call in OS's Kernel and returns status of the system call and any return values.



6. What are the three methods of system call parameter passing?

- Passing parameters in registers

- Parameters stored in a block or table, in memory, and address of block passed as a parameter in a register.



- Parameters placed, or pushed, onto the stack by the program and popped off the stack by the OS.

Good Luck!

New York Institute of Technology
CSCI 330/509-M02 - Operating System
3rd Quiz - Fall 2017
Instructor: Prof. Susan Gass

PART (I) – Multiple Choice Questions:

1. A process control block ____.
A) includes information on the process's state
B) stores the address of the next instruction to be processed by a different process
C) determines which process is to be executed next
D)is an example of a process queue

2. **The ____ of a process contains temporary data such as function parameters, return addresses, and local variables.**
A) text section
B) data section
C) program counter
D) stack

3. The _____ refers to the number of processes in memory.
A) process count
B) long-term scheduler
C) degree of multiprogramming
D) CPU scheduler

4. The list of processes waiting for a particular I/O device is called a(n) ____.
A) standby queue
B) ready queue
C) device queue
D) interrupt queue

5. A blocking `send()` and blocking `receive()` is known as a(n)
A) synchronous message
B) rendezvous
C) blocked message
D) asynchronous message

PART (II) – True/False Questions:

6. The short-term scheduler selects a process that is ready to execute and allocates the CPU to it. T

7. Cascading termination refers to all children processes terminating, while the parent process is still executing. T

8. A mailbox is used in Direct interprocess communication. F

9. In a Non-blocking (asynchronous) interprocess communication, receiver is blocked until a message is available. **F**

10. In a Remote Procedure Call (RPC), a separate stub exists for each separate remote procedure. T

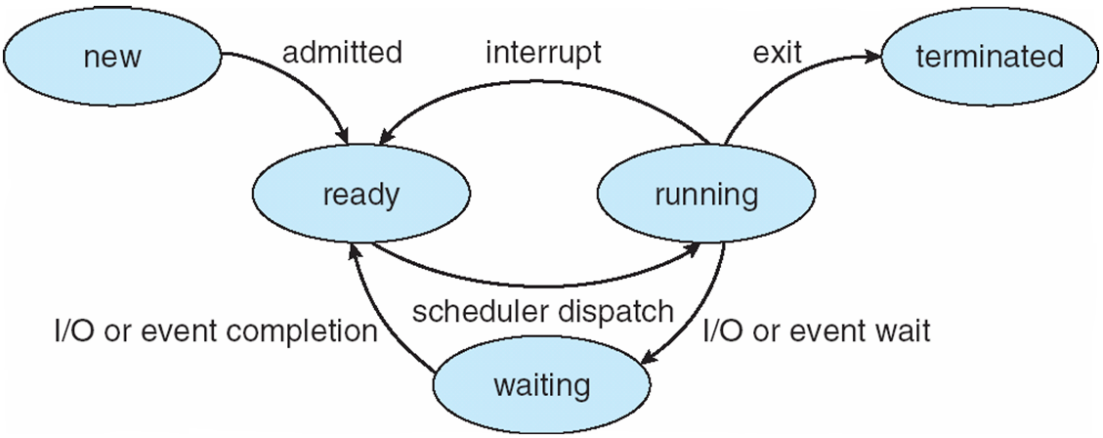
PART (III) – Short Answer Questions:

11. Explain the concept of a context switch.

- Switching CPU from one process to another process, requires the system to **save the state** of the old process (i.e., the one that's being removed from the running state) and load (reload) the **saved state** of the new process via a **context switch**
- **Context** of a process represented in the PCB
- Context-switch time is overhead and idle time for the system

12. Name and describe the different states that a process can exist in at any given time.

- **New** – Process is created
- **Ready** – Processes ready to be selected by CPU Scheduler and assigned to the processor (CPU)
- **Running** – Process is running on CPU
- **Waiting** – Process is waiting for I/O or an event to occur
- **Terminate** – Process is finished execution



Good Luck!

New York Institute of Technology
CSCI 330/509-M02 - Operating System
4rd Quiz - Fall 2017
Instructor: Prof. Susan Gass

PART (I) – Fill-in-the-blank Questions:

- A thread library provides an API for creating and managing threads.
- Cancellation points are associated with deferred cancellation.
- The user-level thread is supported by the thread library.

PART (II) – True/False Questions:

- A thread is a unit of CPU utilization. T
- Each thread has its own register set, stack and data. F (not data)
- LWP is placed between user and kernel threads T
- Task parallelism involves distributing data across multiple computing cores. F (distributes threads NOT data)
- OpenMP provides support for parallel programming in shared-memory environments. T

PART (III) – Short Answer Question:

- List the four major categories of the benefits of multithreaded programming. Briefly explain each.

- **Responsiveness –** may allow continued execution if part of process is blocked, especially important for user interfaces
- **Resource Sharing –** threads share resources of process, easier than shared memory or message passing
- **Economy –** cheaper than process creation, thread switching lower overhead than context switching
- **Scalability –** process can take advantage of multiprocessor architectures

- Briefly describe what a thread pool is and its benefits.

A number of threads are created and placed in a pool where they await work

Advantages:
- Usually slightly faster to service a request with an existing thread than create a new thread
- Allows the number of threads in the application(s) to be bound to the size of the pool.