Quiz 1 - MET CS575 Graduate Operating Systems

Name: Amal Krishna Radhakrishnan Grade: / 15

1. Which of the following statements are true or false about kernel? (1.5 points)

[True or False] Linux kernel code is always executed in kernel mode. That includes dynamic modules such as device drivers. **True**

[True or False] All instructions used in the kernel code are privileged instructions (e.g. only can execute in kernel mode) - **True**

[True or False] Any local variable declared in ISR (Interrupt Service Routine) is stored in the kernel stack. **False**

2. What are the two fundamental models of inter process communication (IPC)? (1 point)

Shared Memory Model - In shared memory model. The cooperating process shares a region of memory for sharing of information. Some operating systems use the supervisor call to create a share memory space. Similarly, some operating system use file system to create RAM disk, which is a virtual disk created in the RAM. The shared files are stored in RAM disk to share the information between processes. The shared files in RAM disk are actually stored in the memory. The Process can share information by writing and reading data to the shared memory location or RAM disk.

Message Passing Model - In this model, data is shared between process by passing and receiving messages between co-operating process. Message passing mechanism is easier to implement than shared memory but it is useful for exchanging smaller amount of data. In message passing mechanism data is exchange between processes through kernel of operating system using system calls. Message passing mechanism is particularly useful in a distributed environment where the communicating processes may reside on different components connected by the network. For example, A data program used on the internet could be designed so that chat participants communicate with each other by exchanging messages. It must be noted that passing message technique is slower than shared memory technique.

3. Describe three general methods for passing parameters to the operating system. (3 points)

a. Pass parameters in registers
b. Registers pass starting addresses of blocks of parameters
c. Parameters can be placed, or pushed, onto the stack by the program, and popped off the stack by the operating system

4. The benefits of multithreaded programming can be broken down into what four major categories? (2 points)

1. **Responsiveness**: Multithreading increases responsiveness i.e. a program with multithreading will remain in the running state even if one of its part is blocked or is executing an extensive task.

2. **Resource sharing**: The threads give the benefit of sharing code and data which allows an app to have a number of different active threads in the same address space.

3. **Economy**: Memory and resource allocation is a costly task for a process being created. But threads share resources of the processes to which they belong, thus they are more cost-effective.

4. **Utilization of multiprocessor architectures**: such architectures are utilized where threads can run, on different processors, in parallel.

5. The diagram in Figure 1 shows the state transition between all five process states. Please specify name and very short description for each state. (2.5 points)

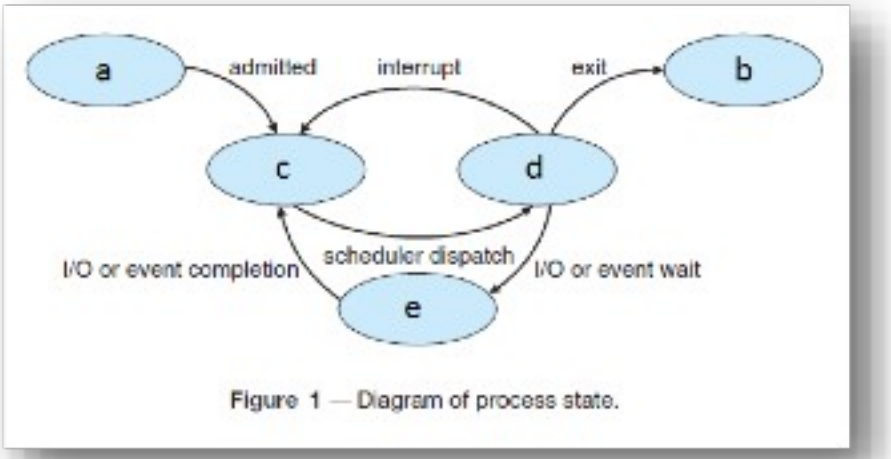

Figure 1 — Diagram of process state.

a. New. The process is being created.

c. Ready. The process is waiting to be assigned to a processor.

d. Running. Instructions are being executed.

e. Terminated. The process has finished execution.

e. **Waiting**. The process is waiting for some event to occur (such as an I/O completion or reception of a signal).

6. Which of the following components of program state are shared across threads in a multithreaded process? Please specify which are shared and which are not shared. (2 points)

Register values

Heap memory

Global variables

Stack memory

Threads share the heap, global variables and the page table. They have private register values and private stack segments. Stack is thread-specific, as well as registers.

7. What are the six types of system calls normally provided by an operating system? (3 points)

Process Control - Processes need to be controlled as in a running process must be able to halt its execution either normally or abnormally. Also, one process may need to run some other process to complete its own execution. So, all these system calls come under this category.

1. end, abort
2. load, execute
3. create process, terminate process
4. get process attributes, set process attributes
5. wait for time, wait event, signal event
6. allocate and free memory

File Management - System calls which deal with operations related to files fall under this type.

1. create file, delete file
2. open, close
3. read, write, reposition
4. get file attributes, set file attributes

Device Management - A process may need several resources for its execution. So system calls used for asking permission from the kernel to use those resources are included in this type.

1. request device, release device
2. read, write, reposition

3. get device attributes, set device attributes
4. logically attach or detach devices

Information Maintenance - We need to keep all the information up to date so these system calls help us to do that.

1. get time or date, set time or date
2. get system data, set system data
3. get process, file, or device attributes
4. set process, file, or device attributes

Communication - Processes need to communicate with each other for many reasons like if they need certain resource which is held by any other process. These system calls assist in doing so.

1. create, delete communication connection
2. send, receive messages
3. transfer status information
4. attach or detach remote devices

Protection - Protection provides a mechanism for controlling access to the resources provided by a computer system. Historically, protection was a concern only on multiprogrammed computer systems with several users. However, with the advent of networking and the Internet, all computer systems, from servers to mobile handheld devices, must be concerned with protection.