**CSC3150 Quiz#1 03/12/2024 Student Name: Student ID:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Section 1** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **Total** |
|  |  |  |  |  |  |  |
| **8** | **9** | **10** | **11** | **12** | **13** | **14** |
|  |  |  |  |  |  |  |
| **15** | **16** | **17** | **18** | **19** | **20** |  |
|  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Section 2** | **1** | **2** | **3** | **4** | **5** | **Total** |
|  |  |  |  |  |
| **6** | **7** | **8** | **9** | **10** |
|  |  |  |  |  |

**Please put your answers of section I and II in the above tables.**

**Section I (40 points): Select one best answer**

1. What is the primary function of an operating system?

D

a) To manage computer hardware resources

b) To provide a user interface

c) To manage software applications

d) All of the above

1. What is the kernel in an operating system?

C

a) The outermost layer of the OS

b) A type of application software

c) The core component that provides basic services to other parts of the OS

d) A tool for managing system resources

D

1. Which of the following is a system service provided by an operating system?

a) File management

b) Memory management

c) Process management

d) All of the above

D

1. System services in an operating system are typically accessed by:

a) System calls

b) Application programming interfaces (APIs)

c) Direct hardware manipulation

d) Both a and b

1. In your assignment 1, task 2, you need to implement IPC between the parent process and the child process by using a memory-mapped file. Which flags must be set for the system call?

C

a. MAP\_SYNC

b. MAP\_PRIVATE

c. MAP\_SHARED

d. MAP\_UNINITIALIZED

6. In your assignment 1,task 1\_1, assuming you already have the number of vertices (v\_cnt) and edges (e\_cnt), what is the minimum number of operations required to load the graph data (vertex, edge, edge\_weight) from the file.

C

a. 1 open(), 3 read()

b. 2 open(), 4 read()

c. 1 open(), 1 read()

d. 2 open(), 2 read()

1. Which of the following is NOT a state in the process life cycle?

D

a) Running

b) Waiting

c) Terminated

d) Sleeping

1. What is a thread in the context of OS?

a) A lightweight process

A

b) A type of system service

c) A hardware component

d) A file on disk

1. How do threads differ from processes?

a) Threads share the same memory space, while processes do not

A

b) Threads are not scheduled by the operating system, while processes are

c) Threads can execute independently, while processes cannot

d) Threads are used for I/O operations, while processes are used for computational tasks

1. What is the benefit of multithreading in an operating system?

C

a) Increased security

b) Reduced memory usage

c) Improved performance through concurrent execution

d) Simplified process management

11.When a process forks a child process, the parent and the child processes could share the same code. So how does the program know whether the current running process is the parent or the child?

B

a) Check the information in the PCB (Process Control Block)

b) Check the pid (returned from the fork), if it is 0, the running process is the child.

c) Check the OS schedule, see which process is scheduled to run

d) Use the **ps -aux** command, the information will be displayed there

1. Which of the following is a characteristic of microkernel-based OS?
   1. Monolithic architecture

B

* 1. High inter-process communication overhead
  2. All system services run in kernel space
  3. Limited scalability

1. What is the advantage of using a microkernel-based OS?
   1. Easier Maintenance and Development

D

* 1. Flexible and scalable
  2. Better isolation and security of system components
  3. All of the above

1. In a microkernel-based OS, where do device drivers typically run?
   1. In kernel space

B

* 1. In user space
  2. On a separate processor
  3. In a virtual machine

1. Which of the following is a challenge associated with microkernel-based OS?
   1. Reduced security

C

* 1. Higher system stability
  2. Increased complexity and overhead from IPC
  3. Higher cost of development

1. Which of the following is a key feature of RPC?

C

* 1. It allows functions to be executed in parallel
  2. It requires manual handling of data serialization
  3. It provides a mechanism for synchronous or asynchronous communication between clients and remote servers
  4. It is primarily used for local communication within the same machine

1. Which of the following is a common use case for RPC?
   1. Data encryption

C

* 1. User authentication
  2. Distributed computing
  3. Local file management

1. What is Advanced Local Procedure Call (ALPC)?
   1. A variant of RPC designed for high-speed network communication

B

* 1. A communication method used for IPC on the same machine
  2. A more structured procedure call mechanism
  3. A security protocol for local networks

1. How does ALPC differ from traditional RPC?
   1. ALPC is slower but more secure

C

* 1. ALPC is used for remote communication, while RPC is for local communication
  2. ALPC is optimized for low-latency communication within the same system
  3. ALPC uses a different serialization format

1. Which one of the following statements is not true for named pipes?
   1. Named pipes are only available on Unix-like OS
   2. Named Pipe are created by calling mkfifo() on Unix.

A

* 1. Named pipes are used for bi-direction communications
  2. Named pipes can be accessed by unrelated processes using a Pipe name

**Section II (40 points): Multiple Choices (zero, one, or more correct answers)**

1) Which of the following are NOT Open Source Operating System?

a) MacOS

A,B,E

b) MS Windows

c) Unix

d) XV6

e) iOS

2) When does the OS need to perform a context switch?

1. In multi-tasking systems, when the OS schedules a different task to the processor.

A,B,C,D

1. In time-sharing system, when a user’s program used up its time slice, and must be preempted.
2. At a system call, the OS must switch from user mode to kernel mode.
3. At an interrupt, the OS must transfer the CPU execution to the interrupt handler.

3) What must be saved at a context switch?

A,B,D

a) The processor states including registers (GPR, FPR, Flags)

b) The PC

c) The stack

d) The memory management information, such as the pointer (e.g.

CR3 in x86) to the page table.

4) Which of the following operating systems are known for using a microkernel or hybrid microkernel architecture?

a) Linux

B,C,D

b) Mach

c) MacOS (Darwin)

d) iOS

b) MS-DOS

5) Which of the following are NVM (Non-Volatile Memory)?

a) SSD

A,B,E

b) HDD

c) DRAM

d) SRAM

e) USB Flash drives

6) If we have a hosted virtual machine running on a processor that supports three modes: root, supervisor, user. The host is a Linux, and two guest virtual machines are a Windows and a Unix. Which of the following statements are true?

a) The host Linux kernel module KVM runs in the root mode.

A,B,C

b) The guest Windows OS runs in supervisor mode

c) All applications in the guest VM run in user mode.

d) All the kernels (Linux, Unix, Windows) run in root mode

7) What is the performance critical parts in an OS?

a) The scheduler

A,B,D

b) The memory manager

c) The compilers

d) The I/O manager

e) The GUI

8) What are the advantages of using Loadable Kernel Modules (LKM) in an

OS?

a) Modularity: LKM can be separately tested and debugged

A,B,C

b) Space Efficient: They are dynamic loadable, if not used, it does not take kernel space

c) They are flexible, especially for devices that are not always in use

d) They tend to increase system performance

e) They will increase system security and reliability

9) Which of the following tools support implicit thread control

a) Intel’s Thread Building Block (TBB)

A,B,C

b) OpenMP library and API

c) APPLE’s GCD (Grand Central Control)

d) POSIX’s **pthread** library

10) In your assignment 1, task 1\_2, what are the motivations for using **mmap**() instead of simply using **read**()?

a) Because the file is too large to fit entirely in memory

A

A,B

b) Because we would like to have fine grained page management

c) Because we need to have Inter-Processes Communication (IPC)

d) Because we need to maintain file content consistency

**Section III (40 points): Short and brief answers**

1. **Kernel vs. System Programs** (8 points)
2. How to distinguish a kernel from a set of system programs shipped with the same OS?

**Kernel: always running, in kernel space, in supervisor mode**

**System program: invoke when needed, in user space, in use mode**

1. Give 4 examples of commonly shipped system programs

**Shell**

**File management tools: rm, mkdir**

**System monitor: ps, top, vmstat**

**Disk utilities: du, fsck, mount**

1. **Interrupts** (8 points)
2. What are the differences between internal exception and external exception?

**Internal:** **Also known as software interrupts, these are triggered by instruction execution (within the CPU). Examples include arithmetic overflow, division by zero, illegal instructions, and system calls,**

**External:** **Also known as hardware interrupts, these are triggered by external hardware devices, such as peripherals or I/O devices, signaling the CPU to handle a specific event. Examples include keyboard input, mouse movements, disk I/O completion, network packets arrival, and timer signals.**

1. Give examples each for internal and external exceptions
2. How are they handled differently?

**Internal exception must be handled immediately**

**External exception can be delayed (queued) a little while**

1. **Process vs. Thread** (8 points)
2. When you do parallel programming, is process based message passing or shared memory based multi-threading more scalable? Why?

**Message passing parallel programming is more scalable.**

**Shared memory architectures are less scalable.**

1. The two primary methods for IPC are shared memory and message passing. If two threads want to communicate with each other, can they also use the same IPC mechanisms?

**YES. They may use message queues for message passing.**

1. **Asynchronized vs Synchronized Communication** (8 points)
2. Give example of asynchronized and synchronized Communications between processes

**Synchronized : MPI\_SEND(), MPI\_recv() Blocking Send/Receive**

**Asynchronized : MPI\_ISEND(), Irecv() Non-blocking Send/Receive**

1. What are the pros and cons of using each communication approach

**Pros: Blocking as a way to synchronize**

**Cons: Lost parallel processing opportunity**

5**) Instrumentation-based vs sampling-based profiles** (8 points)

a) What are the differences between instrument-based vs sampling-based profiling?

**Instrumentation-based profiling is more intrusive, higher overhead, but could be more accurate on collected events**

**Sampling-based profiling is less intrusive, lower overhead, less accurate on collected events.**

1. Give two example tools for each approach that are commonly used.

**Instrumentation-based profiling: Dtrace, Strace, gprof**

**Sampling-based profiling: Perf, Oprofile, Vtune**