Sharar Mahmood (username: sharar)

Attempt 1

Written: Feb 9, 2020 12:32 PM - Feb 12, 2020 5:42 PM

Attempt Feedback

Excellent job on HW1!

Best,
Xiaodong

Submission View

Released: Jan 23, 2020 2:33 PM

Part I. Multiple-Choice Questions: Choose the answer that best applies.

Question 1   2 / 2 points

A _____ uses an existing thread — rather than creating a new one — to complete a task.

○ remote procedure call (RPC)

○ lightweight process

○ scheduler activation

◉ thread pool

Question 2   2 / 2 points

How many arguments does the fork system call take?

◉ 0

○ 1

○ 2

○ 3

Question 3   2 / 2 points

_____ involves distributing tasks across multiple computing cores.

○ Concurrency

◉ Task parallelism

○ Data parallelism

○ Parallelism

Question 4   2 / 2 points

In what way is an operating system like a government?

○ It seldom functions correctly.

◉ It creates an environment within which other programs can do useful work.

○ It performs most useful functions by itself.

○ It is always concerned primarily with the individual's needs.

Question 5   2 / 2 points

What are some other terms for kernel mode?

/

○ supervisor mode

○ monitor mode

○ privileged mode

◉ All of the above

Question 6   2 / 2 points

____ is the number of processes that are completed per time unit.

○ Turnaround time

◉ Throughput

○ CPU utilization

○ Response time

Question 7   2 / 2 points

What statement concerning privileged instructions is considered false?

◉ They cannot be attempted from user mode.

○ They can only be executed in kernel mode.

○ They may cause harm to the system.

○ They change processor modes from kernel to user

Question 8   2 / 2 points

Which symbol denotes a pipe in Linux?

○ >

○ ^

◉ |

○ <

Question 9   2 / 2 points

System calls are said to be expensive because?

◉ they require transition between the user and kernel modes

○ they include library calls

○ computer centers charge a fee for each system call

○ None of the above

Question 10   2 / 2 points

The _____ model maps many user-level threads to one kernel thread.

◉ many-to-one

○ many-to-many

○ one-to-one

○ two-level

Part II. True or False

Question 11   2 / 2 points

A system call is triggered by hardware.

○ True
● False

Question 12	2 / 2 points

There is no universally accepted definition of an operating system.

● True
○ False

Question 13	0 / 2 points

In RR scheduling, the time quantum should be small with respect to the context-switch time.

● True
○ False

Question 14	2 / 2 points

The exec() system call creates a new process.

○ True
● False

Question 15	2 / 2 points

A program and a process are one and the same.

○ True
● False

Question 16	2 / 2 points

System programs run in kernel space.

○ True
● False

Part III. Fill in the appropriate word, phrase or value in the space provided

Question 17	2 / 2 points

There are two types of parallelism: ___data parallelism___ and ___task parallelism___ .

Question 18	5 / 5 points

Many criteria have been suggested to evaluate and compare CPU-scheduling algorithms. The criteria include ___CPU utilization___ , ___Throughput___ , ___Turnaround Time___ , ___Waiting Time___ , and ___Response Time___ .

Question 19	1 / 1 point

In FCFS Scheduling, FCFS stands for ___First Come First Serve___ .

Question 20	2 / 2 points

Cooperating processes require an interprocess communication (IPC) mechanism that will allow them to exchange data and information. There are two fundamental models of interprocess communication: ___Shared Memory___ and ___Message Passing___ .

Question 21	1 / 1 point

Modern operating systems are ___Interrupt___ driven.

Question 22	1 / 1 point

Processes can be described as and divided into two categories, I/O-bound process and ___CPU-bound process___ .

Question 23	1 / 1 point

Linux starts from an initial process with PID 1. It forks child processes, child processes fork child processes. This parent and child relation form a tree, called ___process tree___ .

Question 24	1 / 1 point

Which Register in CPU holds the address of the next instruction to be executed? ___Program Counter (PC)___

Question 25	1 / 1 point

Process switch (process context switch) allocates the CPU from one process to another. It includes two portions: CPU context and ___Storage context___ .

Part IV. Answer all of the following questions.

Question 26	4 / 4 points

Time sharing and multiprogamming are two concepts that are sometimes confused by people who haven't taken CS3110. Describe the difference.

Time sharing allows multiple users to interact at the same time. Emphasizes response time over processor time. It is very important to keep response time short.

In time sharing, the CPU executes multiple jobs by switching among them, but the switches occur so frequently that the users can interact with each program while it is running. It requires an interactive computer system. Multiple users are connected to the central machine via a terminal. The users are served in a round robin fashion creating the isllusion that the user is the sole user of the machine.

Multiprogramming is executing simultaneously mutiple programs on the same processor. OS selects one process and runs it till a wait condition or termination then selects the next process to run. This is advantegous as a different process can be run when one process is waiting for some event such as I/O to occur. Multiprogramming increases CPU utilization by organizing jobs (code and data) so that the CPU always has one to execute.

The main difference is multiprogramming involves running multiple processes on the same processor where as time sharing allows multiple users to interact with the computer simultaneously.

Question 27   5 / 5 points

List 5 types of system resource.

1. The processor (CPU)
2. Memory
3. I/O (input-output) devices
4. File Storage
5. Network connections

Question 28   2 / 2 points

What is confidentiality?

Confidentiality is one of the criteria of system security. It assures that confidential or private information is not made available or disclosed to unauthorized individuals. Confidentiality is one of the requirements of CIA triad.

Question 29   6 / 6 points

Explain the difference between singly, doubly, and circularly linked lists.

Singly Linked List: each item points to its successor. Nodes only have next and NO previous pointers. The tail node next pointer points to null. The direction of traversal is from head to tail only.

Doubly Linked List: a given item can refer either to its predecessor or to its successor. Nodes have next and previous pointers. The head node has null as previous where as tail node has null next. The direction of traversal can be fromhead to tail or tail to head.

Circularly Linked List: the last element in the list refers to the first element, rather than to null. The direction of traversal is from head to tail but since the tail points to the head so there is more flexibility in traversal compared to singly linked list. But it is important to keep track of head and tail as otherwise there is a possibility of running into infinite loops.

Question 30   7 / 7 points

In Unix systems, what system calls have to be executed by a command interpreter or shell in order to start a new process? What do these system calls actually do?

In Unix systems, a fork system call followed by an exec system call need to be performed to start a new process. The fork call clones the currently executing process, while the exec call overlays a new process based on a different executable over the calling process.

Question 31   4 / 4 points

Explain the difference between preemptive and nonpreemptive scheduling.

Preemptive scheduling allows a process to be interrupted in the midst of its execution, taking the CPU away and allocating it to another process. The priority of a new process if greater than the currently running process the new process interrupts the running process and gets CPU time.

Nonpreemptive scheduling ensures that a process relinquishes control of the CPU only when it finishes with its current CPU burst. Suc processes run till termination or waiting conditions.

Question 32   6 / 6 points

Consider the following C code that calls fork(). If you assume that the child process is always scheduled before the parent process, what will be the output?

```
int main()
{
int i;
for (i = 0; i < 3; i++) {
if (fork() == 0) {
printf("Child sees i = %d\n", i);
exit(1);
} else {
printf("Parent sees i = %d\n", i);
}
```

```
    }
}
```
Child sees i = 0
Parent sees i =0
Child sees i = 1
Parent sees i = 1
Child sees i = 2
Parent sees i = 2

▷ View Feedback

Question 33   **3 / 3 points**

Consider the following program. How many times hello is printed? _____
```
#include <stdio.h>
#include <sys/types.h>
int main()
{
fork();
fork();
fork();
printf("hello\n");
return 0;
}
```
Your Answer: 8

Question 34   **2 / 2 points**

What is orphan process?

If a parent process did not invoke wait() and instead terminated, it causes its child process to turn into orphan processes. Linux and UNIX address this scenario by assigning the init process as the new parent to orphan processes. The orphan processes still execute although their parent has terminated.

Question 35   **4 / 4 points**

Describe the similarities and differences of doing a context switch between two processes as compared to doing a context switch between two threads in the same process.

The process context switch needs to save and restore all of the process state, including program counter, registers, memory mapping, accounting, and other resource information.

The thread context switch needs to save the program counter and registers. The memory mapping, accounting, and other resource information stays the same as threads of the same process share that process's resources. Thread context switching consumes less amount of time as compared to process context swtiching.

Question 36   **6 / 6 points**

Suppose that the following processes arrive for execution at the times indicated. Each process will run for the amount of time listed.

Process	Arrival Time	Burst Time
P_1	0.0	7
P_2	2.0	4
P_3	4.0	1
P_4	5.0	4

a) What is the average waiting time for these processes with the SJF scheduling algorithm (or Non-Preemptive SJF)?

___4.0___ b) What is the average waiting time for these processes with the Shortest-Remaining-Time-First scheduling algorithm (or Preemptive SJF)? ___3.0___

Done