**Score: \_\_\_\_\_**

**OSA2 – Multiprogramming:**

**Processes & Context**

**Activities**

COMP256 – Computing Abstractions

Dickinson College

Spring 2023

Prof. Grant Braught

**Name:**

Today’s class introduced the abstractions of multiprogramming and processes. The multiprogramming abstraction allows the user to have multiple programs all running (or at least appearing to run) simultaneously, even if there is only one CPU. The process abstraction groups a program and its context together in a way that allows running programs to be suspended and resumed again later. It is this ability to suspend and resume processes by saving and restoring their context that is used to produce the multiprogramming abstraction. In the activities below you will explore these abstractions a little more, digging first into some of the details of how the OS produces these abstractions and then into a little bit of OS history.

**Multiprogramming via Metaphor:**

🔑 1. Today’s class described three multiprogramming mechanisms and used the college metaphor to help explain them.

a. Give your own explanation of *time sharing* using a sports team as the program and an appropriate athletic facility as the hardware used for running the program.

b. Pick one of the other two multiprogramming mechanisms (*I/O processing time* or *synchronization waits*) and give your own explanation of it using different elements of the college metaphor than were used in class.

🔑 2. In class, a sports team from the college metaphor was used to explain difference between a program and a process. This question asks about a similar explanation using a college class instead of a team.

a. What static information about a college class would be analogous to a program?

b. What additional dynamic information (i.e. context) would a college class have if the “class program” were executing as a process?

🔑 3. In class, a team (program/process) and a coach (OS) were used to explain how a process’ context could be used to suspend and resume a process (i.e. to deep freeze it). In the previous homework you were asked to create your own metaphor including identifying the elements that corresponded to hardware, programs and the OS. This question asks you to apply your metaphor to suspending and restoring a process.

a. Reintroduce your metaphor in one sentence. Be sure you clearly identify the elements playing the hardware, programs and operating system in your metaphor. (E.g. I would say, my metaphor is that of a college with the campus facilities being the hardware, students, classes and teams being programs and faculty, staff and coaches being the OS.)

b. State the specific individual elements of your metaphor that will play the role of the program and the OS for this question. (E.g. I would say, a sports team is playing the role of the program and the coach is playing the role of the OS.)

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | **System Component** | **Metaphor Element** |  |
|  | Hardware |  |  |
|  | User Programs |  |  |
|  | Operating System |  |  |
|  |  |  |  |

c. When the program you identified in part b is running, what context will the resulting process have? Your answer should make it clear that the context you describe is reasonably sufficient to suspend and restart the process.

**Hardware for Multiprogramming:**

🔑 4. The offset register was introduced as hardware support for multiprogramming to allow programs to be moved to different parts of the computer’s memory.

a. For a given process, what does the value in the offset register indicate?

b. Imagine that the offset register for a process contains the value 2060.

i. If that program executes a JUMP instruction that jumps to logical address 100 in the program, what value will be in the program counter (PC) after the JUMP instruction?

ii. From what physical memory address will the next instruction be fetched?

**Context Switching:**

🔑 5. What is the name of and abbreviation for the data structure into which a process’ context is preserved/saved?

🔑 6. Context switches are an essential part of multiprogramming.

a. Briefly describe in your own words what happens during a context switch.

b. Briefly describe in your own words how context switching makes multiprogramming possible.

🏆 7. When implementing function calls in assembly language, we preserve only the contents of the general purpose registers (R0-R11) that were used by the function and (if necessary) the return address (R12). But we did not preserve the special purpose registers: the stack pointer (R13) or the return value (R14) or the scratch register (R15).

However, the OS must preserve/save all registers as a part of the process’ context. Briefly explain why all registers must be preserved/saved and restored during a context switch.

**Process States:**

It is not required viewing, but if you would like another explanation of the process states that were introduced in class you can watch the video *Process Life Cycle: States* by Ada Gavrilovska from her Udacity course on operating systems.

* <https://www.youtube.com/watch?v=DacMRZRc4Bo> (2:29)

🔑 8. Identify the five process states. For each state, give a one sentence explanation of why or when a process will be in that state.

9. What are the main things that the operating system does when a process is in the new state?

10. As the OS moves processes in a multiprogramming system between the different states the OS must perform context switches. These switches save the context of the process being stopped to its PCB and restore the context of the process being started from its PCB.

The arrows in the Process States diagram from class indicate the ways that an OS might move a process’ PCB from one state to another. Each of these arrows has a label. That label is the name of the *transition* from one state to another. For example, the transition from the “New” state to the “Ready” state is called “Created.” Use the Process state diagram and the names of the transitions to answer the following questions.

a. Which transition(s) in the diagram will require that the OS save the state of a process into its PCB?

b. Which transition(s) in the diagram require that the OS restore the state of a process from its PCB to the CPU?

🔑 11. In a few sentences of your own words, describe the job of the operating system’s scheduling routine. Be sure to mention the Ready and Running states in your explanation.

**Interrupts:**

🔑 12. Recall that *hardware interrupts* (or just *interrupts*) are signals from hardware devices that cause code in the OS to run. When an interrupt occurs, the OS code that is run will move process PCB between the different states as necessary to keep track of what is happening in the system. Briefly describe how the OS would move the PCBs between states in response to each of the following interrupts:

a. Assume that there is a PCB in the running state and several PCBs in the Ready state. The timer device generates a hardware interrupt indicating that the running process’ turn to run (i.e. its *time slice*) has expired.

b. Assume that there is a PCB in the running state and that the PCB for another process is in the Waiting state. The process in the Waiting state had been moved there because it had previously requested some data from the hard disk drive. The hard disk drive generates a hardware interrupt indicating that the data requested has been successfully read from the disk and is ready for processing.

**A Little OS History:**

Watch the first 5:00 minutes of *Operating Systems* from the Crash Course series with Carrie Anne. The rest of the video touches a variety of OS topics that we’ll also be covering in the coming classes. So, if you are enjoying it and want a preview, feel free to keep watching.

* <https://www.youtube.com/watch?v=26QPDBe-NB8> (First 5:00 of 13:35 total)

13. The video says that operating systems were “born” because we needed a way for computers to operate themselves. Briefly explain in your own words why it was necessary for computers to operate themselves.

14. In a sentence or two of your own words, briefly explain what batch processing is.

15. In a sentence or two of your own words, briefly explain what device drivers are and how, as a part of the OS, they simplified the job of programmers.

16. The Atlas Supervisor was an early operating system. What was the big advance made by the Atlas Supervisor? Why was that advance necessary?

17. What term does Carrie Anne use as a synonym for multiprogramming?

Optional: To help me improve and scope these activities for future semesters please consider providing the following feedback.

a. Approximately how much time did you spend on this activity outside of class time?

b. Please comment on any particular challenges you faced in completing this activity.