

CSE-381: Systems 2

Homework #3 Part A

Due: Wed Sept. 18 2019 before 11:59 PM (Midnight)

Email-based help Cutoff: 5:00 PM on Tue, Sept 17 2019

Maximum Points for This Part: 18

Submission Instructions

This homework assignment must be turned-in electronically via Canvas. Type in your responses to each question (right after the question) in this MS-Word document. You may use as much space as you need to respond to a given question. Once you have completed the assignment, **save it as a PDF file, and upload the PDF** to Canvas.

Note that copy-pasting from electronic resources is plagiarism. Consequently, you must suitably paraphrase the material in your own words when answering the following questions.

Name: Craig Beuerlein

Objective

The objective of this homework is to review and recapitulate the basic concepts related to operating systems and C++ programming.

Review the Chapter #1 and Chapter #2 from the reference book “Operating System Concepts” (**Link available off Syllabus page on Canvas**) before proceeding with this exercise.

Mandatory videos to study -- you will get questions in exams based on the content in these videos:

1. Introduction to OS: <https://youtu.be/PPSmAlnrJxo>
2. Booting: <https://youtu.be/7D4qiFIosWk>

1. Briefly describe (with at least 4 to 5 sentences for each) the two major functions of an operating system using suitable examples for each one of the scenarios (watch first video) **[2 points]**

- a. Function #1 (with 2 examples):

An operating system manages computer resources of the system. A program may need to send commands to the CPU to be processed so an OS will interact with the program to send data to the CPU. A program may need to send memory to RAM, so an OS will assign a memory address for a program to store data.

- b. Function #2 (with 2 examples):

An operating system acts as an intermediary between the computer hardware and the user. An OS will manage all commands of the user to prevent errors that a user might perform. If a user is using any I/O devices, the OS will manage and control the operations performed through the I/O devices.

2. The following questions relate to Boot Loaders (from second video) **[2 points]**

- a. What is a boot loader and what is its primary purpose?

An OS specific program that loads the OS into the systems memory.

- b. Where is a boot loader found on a permanent storage device such as a floppy disk?

The bootloader is stored in a partition in a permanent storage device called the boot sector.

- c. What is a 2-stage boot loader?

A 2-stage bootloader consists of 2 programs, a master boot record when run starts the OS bootloader to properly load the OS.

- d. Why is 2-stage boot loading used?

With larger operating systems, the size constraints of the master bootloader may present all the necessary functions from starting an OS to be included. An OS bootloader is used to fully launch the OS.

3. What is a system call and how does it operate? **[1 point]**

A system call interacts with OS kernel to perform a function. They are used to develop software systems. When a program runs it interacts with a system call to interact with the kernel.

4. Describe three general methods for passing parameters to the operating system in a system call (**exactly the same as question 2.13 from textbook so read the relevant section in the textbook**) **[2 points]**

- 1 A system call can be used to pass parameters from a memory registry to the OS
- 2 Parameters stored in a stack can be pushed and popped from the stack to the OS

3 Parameters can be stored in blocks and passed to the OS by sending the memory register of the block of parameters

5. From a Linux perspective, briefly (2 sentences) describe the use of user id (a number) and group id (a number) [1 points]

User ids are assigned to every user in a linux system and determine what resources that user has access to. A group id is used to grant access to resources to multiple users in Linux.

6. Table at least 2 significant differences between batch processing and multiprocessing. [1 points]

<i>Batch Processing</i>	<i>Multiprocessing</i>
Batch processing performs tasks one after another and therefore is slower.	Multiprocessing allows many tasks to be completed at once and therefore is faster.
CPU may idle	CPU never idles

7. Briefly discuss a realistic scenario when designing an operating system as a virtual machine would be advantageous. [1 points]

If a computer is running multiple programs at once and begins to run out of storage, an OS can move data from RAM to storage that a program that is not being used at the present time needs to continue to function.

8. What is virtual memory? Why is it used on most modern operating systems? [1 points]

Virtual memory is used to supplement RAM by transferring data from RAM to physical storage. OSs use virtual memory to run more programs that require more data than a computer has RAM to hold.

9. What is the difference between a Type-1 and a Type-2 hypervisor? [1 points]

A Type-1 hypervisor runs directly on the computer's hardware without the need for an OS to interact with the hardware. A Type-2 hypervisor uses an OS to interact with the computer's hardware.

Review programming from CSE-278

The objective of the next few questions is to review pertinent programming concepts from CSE-278

Prior to answering the questions in this homework it may be beneficial to briefly following chapters from the E-book titled “[C++ How to Program](#)” ([Link to e-textbook is in the Syllabus on Canvas](#)):

- Chapter 1 (Introduction)
- Chapter 2 (Expressions & I/O)
- Chapter 3.1, 3.2 (String & methods)
- Chapter 4 (if-statements)
- Chapter 5.1 – 5.10 (loops & switch)
- Chapter 6 (functions/methods)

10. What is path to a file? Using suitable examples explain the differences between relative and absolute path? (answer in 2 or 3 separate sentences) [1 point]

A file path is the unique location of file in a file system. An absolute file path is the file path from the root of the drive to the file location. A relative file bath is the path from the current directory to the file location.

11. Briefly (2 to 3 sentences each) discuss pass-by-value versus pass-by-reference mechanisms for passing parameters to methods. As per this course's conventions, which [approach is preferred based on data type](#) of parameters in C++? [2 points]

A pass-by-value takes the value from the memory location and any changes made to the value do not effect the value stored in the memory location. A pass-by-reference manipulates the value stored at the memory location. Pass-by-values are preferred for parameters

12. List at least 3 unique properties that can be inferred from data type of a variable? [1 point]

1. What type of information is stored within the variable
2. Size of the datatype
3. What the variable can be used for

13. What is the `size_type` or `size_t` data type? Illustrate its use with an [example of a for-loop](#) (do not write whole program, but [just 1 line](#) of the for-loop) [1 point]

Example:

14. Given a constant string object `str` (`const std::string str`) what methods can be called on `str`? Give an example of 1 method that **can** and 1 method that **cannot** be called on the object `str`? [1 point]

- (a) Can call:
- (b) Cannot call: