

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: Ce Zhang

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/7D4qiFlosWk>

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):

This is the important function of an OS is to act as a resource manager. resources include main memory, disk space, screen space, CPU time, There are some example:

1. The OS will have switch I/O process to manage CPU time.
2. The OS also prevent specific programs from accessing devices which can help os to manage them together.

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

One of the functions of an OS , some examples:

- For example, an OS may stimulate the machine which cannot show exist.
- It may provide additional virtual memory which look like their will be other physical memory.

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?

A boot loader is a type of program that loads and starts the boot time tasks and processes of an operating system or the computer system. Its primary purpose is loading an OS into memory from.

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

The boot loader is stored in logical sector 0, and it is called the boot sector. The Windows OS put his boot code in the first sector and storage of information in RAM memory needs to be stored permanently in the floppy disk.

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

Both them is 2 boot loaders which can load an OS. The first stage is simple second stage boot loader version and it will be loaded also. The second stage loader will be very complicated and this os will loads to the memory.

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

Boot sectors are 512bytes. This size is really small. And which will let second section loads.

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

User application and interacts OS is called a system call. When it calls, the machine is switch from user-space to kernel-space and the system callwill hppen in 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. Passing via CPU registers – this is very common and used in a lot of machine, and it is also the fastest way.
2. Pass by reference – this is stored in memory location and the address of memory is passed by the register.
3. Passing values on the stack –this is used by the some operating system and do not have a limit in the number and length 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]

A user id is a unique positive integer which is assigned by the Unix. Each user is identified and unique to the system by their UID, and user names are used only as an interface.

Uid = 10000 gid =100, groups = 100

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

| <i>Batch Processing</i> | <i>Multiprocessing</i> |
|---|---|
| One program can be run in any time | Multiple programs can be run in the same time |
| One program has all resources on the machine. | Multiple can use and share the resource for each other. |
| | |

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

When the batch processing applications are ported to newer hardware systems, then designing the OS will be virtual machine that provides whole backwards.

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

Virtual memory is a memory management capability of an operating system that will uses hardware and software to allow a computer to use for physical memory shortages by temporarily transferring data from RAM to disk machine.

Because it need very fast speed and very efficiently.

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

Type-1 can runs directly on the hardware and there is no host operating system. Type-2 hypervisor runs on top of a host operating system and guest should use operating system runs on the virtual memory which is created by the Type-2.

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]

An absolute path refers to the same location in a file system relative to the root directory.

Example:

Absolute:

\root\desktop\hw1\hw12.cpp

Relative:

...\hw1\hw12.cpp

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]

Passing by reference means the called functions' parameter which is the same as the callers' passed argument.

Pass by value means the called functions' parameter which is a copy of the callers' passed argument.

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

1.range of valid values

2.predered method for passing

3.minimum storage required to store.

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]

The `size_type` or `size_t` data type is simply an alias for unsigned long.
Example: `for((size_t) x, x< number, x++)`

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: the method which do not need to change the string can be call
- (b) Cannot call: the method which should change the string cannot call.