CMPS 2301 Introduction to Computer Systems & Networking

Lab 7: OS Summary

Vaughn Bangash, Grayson Buchholz, Sam Taylor, Stanley Nguyen

Objectives

- To re-examine and reflect on ideas of the Operating Systems part of this course

You may complete this lab in groups of three or four.

1 Getting Started

We have so far seen a great number of issues that occur in design of operating systems, and computer systems in general. One of the goals of our course is to leave you with

- an understanding of problems that a computer system needs to solve and of possible solutions to those problems;
- and an ability to identify similar problems and apply similar strategic solutions in other subject areas.

In this assignment your task will be to look back at the first part of the course and summarize the interesting ideas that you encountered. You'll be looking for new terms that you learned and the problems they describe, and notable problem-solving approaches, solutions, hacks and design tricks.

Read the entire assignment before you begin working on it.

Lab 7: OS Summary

2 Assignment

Identify and summarize these issues in the following tables. Some cells have already been filled to serve as an example.

1. Interesting terms and concepts (aka new fancy words you learned or revisited in this course)

Term and/or brief summary	Computer system examples	Examples from different academic areas (including other branches of CS)	Examples from everyday life
Caching (storing a copy of recently accessed data in a storage that's faster than the primary storage).	- Processor cache maintains small subset of data from main memory. - Flash drive is used as cache for HDD.	Human brain studying material, retaining only some of it to where it can be immediately recalled	Sticky notes, summarizing crucial data with ease of access
Privileged access / execution	-Trapping into kernel mode where RAM can be written to / read from directly	In cybersecurity, the various OS privilege levels (Unprivileged & Administrator & System on windows, root on unix)	Access key cards in an office building
Scheduling of tasks / processes	-Muti-level-feedback queue -round robin, -Shortest runtime	Priorities in governance (Poli Sci)	Making a schedule for your busy day
Virtualization	-CPU Virtualization (Threads, processes) -Memory virtualization (Paging)	-Virtual machines -Software emulation	Using a VPN/Remote Desktop to use the functions of your office work PC from your home laptop
Address translation	-Virtual to Physical address (VPN to PFN)	Cipher encryption, Language translations	Becoming a server host to allow others to simultaneously interact on your network
Policies	-Scheduling policies, Replacement policy, (OS algorithms for how the computer should handle certain situations)	Plugging in numbers for equations used to represent laws in physics. (The laws tell you what to do in that situation)	Legal or ethical codes that tell you how to respond in certain situations
Process	-Any running program	A computer security software running a single security scan on a user's hard dive	-Word processors -Calculators

CMPS 2301 Introduction to Computer Systems & Networking

Lab 7: OS Summary

Limited Direct Execution	-Time sharing of the CPU by running different processes at different times -CPU scheduling	Deadlock prevention algorithms, prevent one process from getting stuck by running processes at different times	Computing systems used in banking when a user swipes debit card, one process for initial purchase, process for money transfer, and one process for end user to receive

2. Useful problem-solving approaches

Brief statement	**Computer system examples**	**Examples from different academic areas (including other branches of CS)**	**Examples from everyday life**
Periodically check the system status in order to detect errors early.	-Device polling (I/O, drivers)	Periodically check the status of a biological experiment to catch obvious problems early.	Checking your rearview mirror periodically so you know what lane's clear when you want to change lanes
Abstraction	-Calling a system call instead of attempting all of its CPU instructions directly & at once	Software displays its basic functionalities without going in-depth of every line of code	Explaining the basic idea of a concept without getting into complex detail
Assigning automatic responses to troubleshoot an issue	-Interrupt handlers, page fault handlers	Programming a robot to move backwards and reorient after hitting a wall	Sensors in a car detect engine RPMs redlining and know to downshift
Context Switching	Kernel and User Mode switches, Interrupt handlers	Saving one document before working on another	Pulling over to send a text or make a phone call before continuing to drive.
Segmentation	Code, heap, stack, free; Address translations/Paging;	Dividing a math problem into sections, the base case and the proof case.	Dividing your day into three sections based on meals; breakfast in the morning, lunch in the afternoon, and dinner at night.

Lab 7: OS Summary

Mechanisms	Used by the OS to automatically complete guidelines of policies	When typing an equation into a calculator, the calculator is your mechanism and the equation is your policy.	When running, your legs are your mechanisms, and your brain telling your body which direction to go serves as the policy.
Threads	Utilizing multiple points of execution permits context switching without changing pages, creating some shared memory	Two graphs, with differing domains and ranges, intersecting at the same point.	A pencil can either write or erase, but cannot do both at the same time.
Paging	Dividing a process into multiple memory segments so they can be stored in separate places	Saving parts of the same document in different places to be combined later	Creating a schedule on a calendar to divide up the workload.
Swapping	Used in the cache to minimize the number of times a page has to be fetched from disk	An equation sheet for math allows for easy access to commonly used formulas.	Making sure you have everything you need for the day so you don't have to go back home to get it

3. Neat solutions, hacks, design tricks

Brief statement	**Computer system examples**	**Examples from different academic areas (including other branches of CS)**	**Examples from everyday life**
Devising separate solutions for initial system startup, system resume, and normal system operation.	Different types of actions are performed at OS booting, standard OS operation, and OS shutdown.	Computer Task Manager having memory priority to tasks immediately needing more attention	Similarly, different types of actions are performed at starting a car, continuous running of a car, and car shutoff.
Reserving a resource during the completion of a particular task	-Locks -Setting all other process's states to blocked/ready to monopolize CPU	In long division the answer is slowly found one digit at a time, carrying the earlier digits to form the final answer.	Holding a hot pocket while microwaving the other one.

Utilizing different strategies/priorities to maximize efficiency	-FIFO, LRU, SJF, STCF, etc.	Completing assignments based on due date, difficulty, and work-load left	Brushing your teeth in the shower to minimize time getting up in the morning.
Creating mutually exclusive areas for critical sections	Locks/concurrency	The function of the mitochondria in biology.	A car key ensures that only one person can use the vehicle at a time and determines who unlock and drive the car.
Utilizing system calls so programs can access operating system functions	-Fork(), Wait(), Exec(), etc	API calls that utilize functionality of systems directly while providing a user-friendly interface for doing so	3d Rendering kits only calling functions when object needs to be rendered, through an Exec() function. All non-rendered parts will be under Wait() until needed
Using timed checks to balance prioritization within a system	Timer Interrupt	Computer systems automatically doing periodic audits to update software	At some certain interval, your car checks tire pressure to know if you're getting flat
Divide different system functions based on purpose and handling	Separation of RAM, HDD, CPU, GPU, etc (Von Neuman Architecture)		

4. Other (optional, bonus points)

If there are other interesting ideas from the course that don't fit into the above categories, name them and provide examples here. Essentially, you're looking to create a cheat-sheet that you could use as a reference for most essential course content in the future.

Quality criteria

Your goal is to come up with as many entries as possible based on the material in all assigned reading chapters from OS book. The bare (passing) minimum is total of 10 new added rows among all three tables; the ideal is 20 new entries or more. It may not be possible to come up with three good examples to complete a row (computer system example, other academic area, everyday life), so it's ok to leave a cell or two in a row unfilled (although you should attempt to fill all of them). You should feel free to add several examples in one cell.

When to stop: if a table follows my formatting (11 pt font with standard spacing) and is beginning to exceed one page in length, stop working on that table.

CMPS 2301 Introduction to Computer Systems & Networking

Lab 7: OS Summary

If you have doubts whether your example is falling into academic area (e.g. medicine) or everyday life (e.g. health issues) pick either of them. Similarly, if you're not sure whether your idea fits better into table 2 or 3, pick either. In general, when you're not sure about some step of the assignment, you are encouraged to take the decision that makes most sense to you, and briefly explain it, rather than stare at it without making progress.

Collaboration guidelines

You shall work together in groups of 3 or 4 and submit one document for all. All members of the group:

- should understand and be able to explain and defend all info put in the document;
- will receive the same grade for this assignment;
- should submit the same document featuring the names of all group members.

Copying guidelines

You are allowed to use any hardcopy or electronic resource that's necessary to help you complete this assignment under the following three guidelines:

- There should not be copy-and-pasting involved. You can include ideas from external resources but all text in the assignment should be typed by you (or members of your group).
- You need to include the references: links for electronic resources, bibliographic information for books, etc. No need to provide bibliographic information for your textbook.
- The resource should not forbid such use (for example, if the website says that you cannot use its materials, then you cannot use them).

Two important comments:

I plan to make your answers to this exercise available to class. If you copy-and-paste, copy without proper attribution, etc, this will become visible to everyone. The flip side is that your great answers will also be visible to everyone.

You need to approach all external material analytically, and fully and firmly believe in its correctness when you include it in your submission. If you don't, you'll be trusting your lab grade to strangers, not a great idea.

Elements of style

Lab 7: OS Summary

Check your grammar, punctuation, spelling. Avoid using filler whitespace and filler text.

Submission

Submit your work on Canvas as a .pdf document containing your names, three tables, references section. Optionally, include a section with any other relevant comments on OS portion of the course.