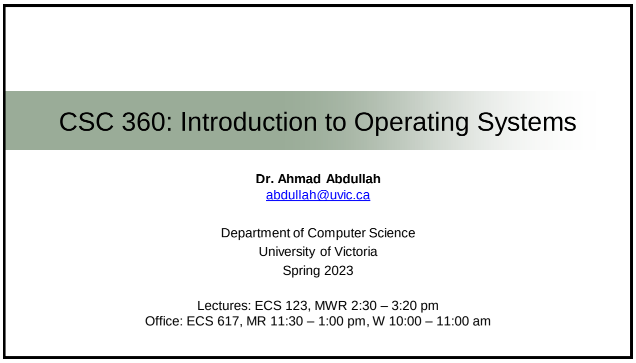


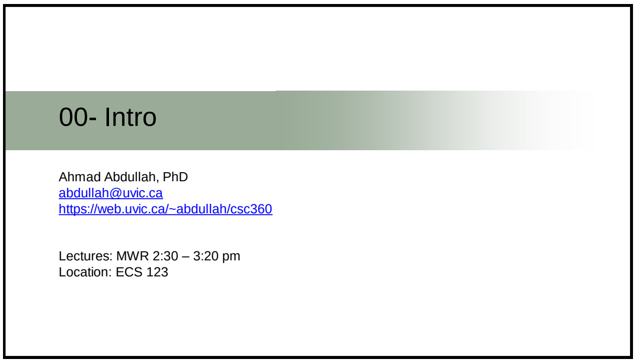
CSC 360: Introduction to Operating Systems

Dr. Ahmad Abdullah
abdullah@uvic.ca

Department of Computer Science
University of Victoria
Spring 2023

Lectures: ECS 123, MWR 2:30 – 3:20 pm
Office: ECS 617, MR 11:30 – 1:00 pm, W 10:00 – 11:00 am

1

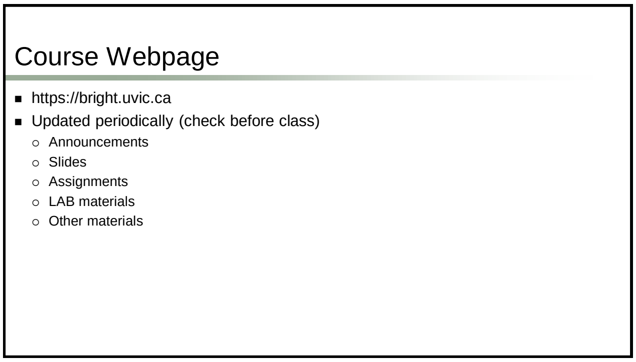


00- Intro

Ahmad Abdullah, PhD
abdullah@uvic.ca
https://web.uvic.ca/~abdullah/csc360

Lectures: MWR 2:30 – 3:20 pm
Location: ECS 123

2



Course Webpage

- https://bright.uvic.ca
- Updated periodically (check before class)
  - Announcements
  - Slides
  - Assignments
  - LAB materials
  - Other materials

3

Introduction to Operating Systems

- This course aims to provide an introduction to major concepts of operating systems and study of the interrelationships between the operating system and the architecture of computer systems. Topics discussed include operating system structures, concurrent programming techniques, cpu scheduling, deadlocks, memory management, file systems and protection.

4

What this course covers?

- The purpose of an operating system
- Influence of computer-system organization and computer-system architecture
- Typical structures for an OS (including a kernel)
- Services and operations provided by an OS
- The process concept
- Support for concurrency (synchronization, communication)
- Memory management from the perspective of processes and OS services
- Management of different types of data storage
- Security

5

Textbook

- Title:
  - “Operating Systems Concepts”, 10th edition
- Authors:
  - Silberschatz, Galvin, and Gagne
- Publisher:
  - Wiley

6

Tutorials meant to help with assignments

- Tutorials will be used where necessary
  - This is unlike a lab-based course where there are labs every week
- Tutorials will be announced the week before they are to take place
  - Please attend your tutorial section for that week
  - Working plan right now is for the first tutorial to take place the week of Jan 23.

7

Evaluation

- Official course outline:
  - https://heat.csc.uvic.ca/coview/course/2023011/CSC360
  - Link to outline also available via Brightspace site
- Three programming assignments in C (15% each)
- Midterm exams (Thursday, February 16th; 20%)
- Final exam (scheduled by UVic admin, 35%)
- Please read the outline ASAP!
- Link to outline is provided at the Brightspace site for our course

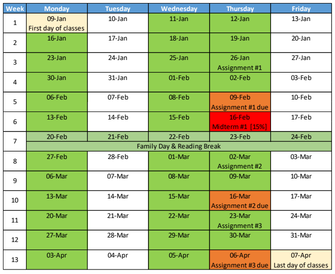
8

Evaluation

Assignment/Exam	Weight	Assigned Date	Due Date
Assignment 1	15%	26-Jan	09-Feb
Midterm 1	20%	16-Feb	-
Assignment 2	15%	02-Mar	16-Mar
Assignment 3	15%	23-Mar	06-Apr
Final Exam	35%	TBD	-

9

SPRING 2023 Calendar



Week	Monday	Tuesday	Wednesday	Thursday	Friday
1	09-Jan First day of classes	10-Jan	11-Jan	12-Jan	13-Jan
2	16-Jan	17-Jan	18-Jan	19-Jan	20-Jan
3	23-Jan	24-Jan	25-Jan	26-Jan Assignment #1	27-Jan
4	30-Jan	31-Jan	01-Feb	02-Feb	03-Feb
5	06-Feb	07-Feb	08-Feb	09-Feb Assignment #1 due	10-Feb
6	13-Feb	14-Feb	15-Feb	16-Feb Midterm #1 [15%]	17-Feb
7	20-Feb Family Day & Reading Break	21-Feb	22-Feb	23-Feb	24-Feb
8	27-Feb	28-Feb	01-Mar	02-Mar Assignment #2	03-Mar
9	06-Mar	07-Mar	08-Mar	09-Mar	10-Mar
10	13-Mar	14-Mar	15-Mar	16-Mar Assignment #2 due	17-Mar
11	20-Mar	21-Mar	22-Mar	23-Mar Assignment #3	24-Mar
12	27-Mar	28-Mar	29-Mar	30-Mar	31-Mar
13	03-Apr	04-Apr	05-Apr	06-Apr Assignment #3 due	07-Apr Last day of classes

10

Credits

- Special thanks to Professors Kui Wu and Mike Zastre for letting me use their course material.

Thank you!

11

Class Interaction

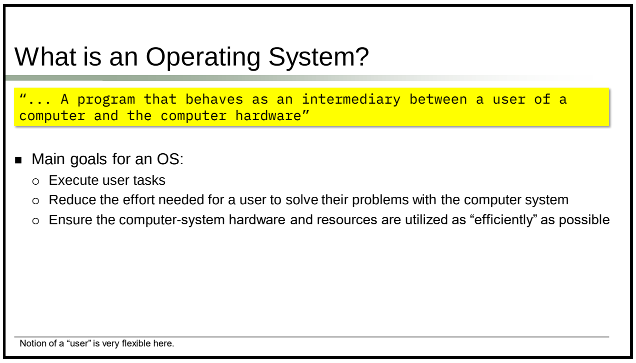
- We will be using www.menti.com a little bit during lectures this semester
  - Informal polls
  - Non-graded quizzes and questions
  - Anonymous
  - None of this on menti.comis for course credit!
- Link and code to Mentimeter will be shown when there is something to do.
- You do not need an account with Mentimeter to complete any of this.
  - Ignore everything at www.menti.comasking you to sign up for an account!

12

What is an Operating System?

https://www.menti.com/alwt1mf5dcq9

13



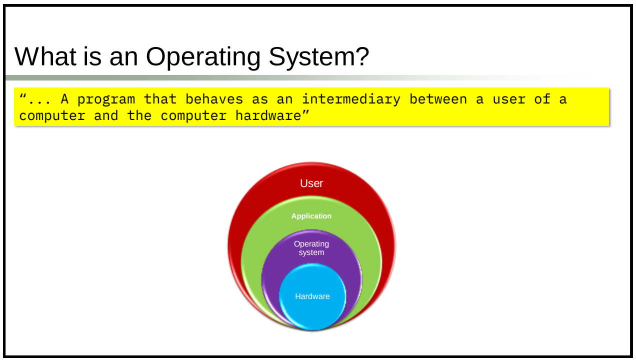
What is an Operating System?

“... A program that behaves as an intermediary between a user of a computer and the computer hardware”

- Main goals for an OS:
  - Execute user tasks
  - Reduce the effort needed for a user to solve their problems with the computer system
  - Ensure the computer-system hardware and resources are utilized as “efficiently” as possible

Notion of a “user” is very flexible here.

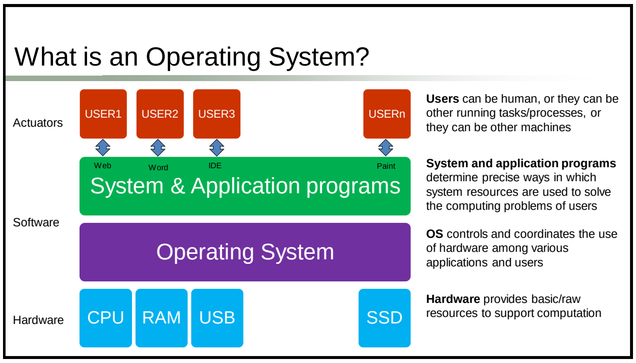
14



What is an Operating System?

“... A program that behaves as an intermediary between a user of a computer and the computer hardware”

15



What is an Operating System?

Actuators

USER1 USER2 USER3 USERn

Web Word IDE Paint

Software

System & Application programs

Operating System

Hardware

CPU RAM USB SSD

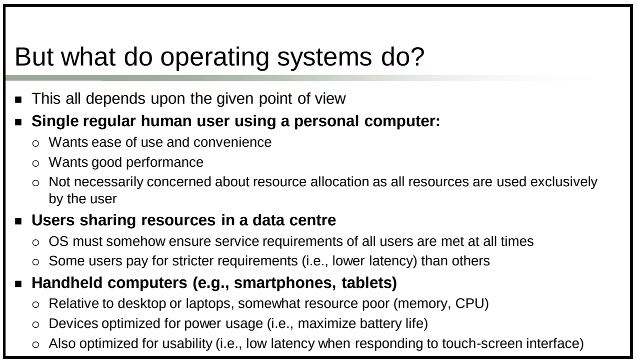
Users can be human, or they can be other running tasks/processes, or they can be other machines

System and application programs determine precise ways in which system resources are used to solve the computing problems of users

OS controls and coordinates the use of hardware among various applications and users

Hardware provides basic/raw resources to support computation

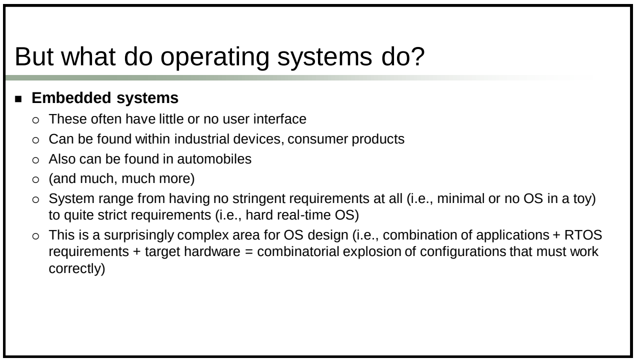
16



But what do operating systems do?

- This all depends upon the given point of view
- **Single regular human user using a personal computer:**
  - Wants ease of use and convenience
  - Wants good performance
  - Not necessarily concerned about resource allocation as all resources are used exclusively by the user
- **Users sharing resources in a data centre**
  - OS must somehow ensure service requirements of all users are met at all times
  - Some users pay for stricter requirements (i.e., lower latency) than others
- **Handheld computers (e.g., smartphones, tablets)**
  - Relative to desktop or laptops, somewhat resource poor (memory, CPU)
  - Devices optimized for power usage (i.e., maximize battery life)
  - Also optimized for usability (i.e., low latency when responding to touch-screen interface)

17



But what do operating systems do?

- **Embedded systems**
  - These often have little or no user interface
  - Can be found within industrial devices, consumer products
  - Also can be found in automobiles
  - (and much, much more)
  - System range from having no stringent requirements at all (i.e., minimal or no OS in a toy) to quite strict requirements (i.e., hard real-time OS)
  - This is a surprisingly complex area for OS design (i.e., combination of applications + RTOS requirements + target hardware = combinatorial explosion of configurations that must work correctly)

18

An operating system is...

- **... a resource allocator:**
  - Manages nearly all computational, memory, storage devices
  - Arbitrates between conflicting requests for resources
  - Must somehow ensure efficient and fair use of resources
- **... a control program:**
  - Starts and stops user programs
  - Ensures errors in execution do not result in crashing system
  - Must also detect and prevent improper use of resources (i.e., access control for filesystem items)

19

An operating system is...

- **... surprisingly hard to define!**
  - No definition is accepted universally
- **Some have (somewhat ironically) defined it as “Everything a vendor ships when you order an OS.”**
- **Others have (also somewhat ironically) defined it as “The one program that runs at all times on the computer” (i.e., the kernel)**
- **Everything else is either:**
  - A system program(such as a compiler, assembler, etc.) that ships with the OS, or...
  - ... is an application program.

20

A bit of history...

Fernando Corbato (1926-2019)

Won the ACM Turing Award in 1990 for his work on operating systems (Multics, CTSS, etc.)

https://youtu.be/Q07PhW5sCEk

21

A bit of history...

Margaret Hamilton (b. 1936)

https://youtu.be/4sKY6_nBLG0

Member of team the wrote the on-board flight software from the NASA Apollo program. She is also one of the creators of the term “software engineering”.

22

Some aspects of OS operation

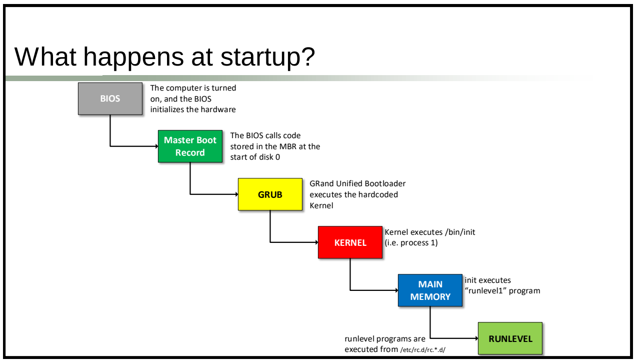
- Given that a top-down definition is out of reach...
- **... we will instead approach our study of operating systems by looking at them from different perspectives.**
- Some initial perspectives (i.e., not an exhaustive list):
  - What happens at startup?
  - What are the consequences of physical concurrency?
  - What are the consequences of different data storage devices (speeds, capacities)?
  - What are some typical OS structures?
  - What are foundational OS abstractions?

23

What happens at startup?

- **Bootstrap program**
  - Minimal program loaded into computer at power-up or reboot
  - Normally stored in ROM or EPROM (aka firmware)
- These programs load in progressively more powerful programs
  - Initializes all aspects of the computer system
  - Final step is to load OS kernel and begin its execution (i.e., in Unix, start process 0, which forks itself to create in process 1, such that this latter process is the ancestor of all other processes).
  - Note: We will go into gruesome detail this semester on the concept of a **process**

24



What happens at startup?

BIOS

The computer is turned on, and the BIOS initializes the hardware

Master Boot Record

The BIOS calls code stored in the MBR at the start of disk 0

GRUB

GRand Unified Bootloader executes the hardcoded Kernel

KERNEL

Kernel executes /bin/init (i.e. process 1)

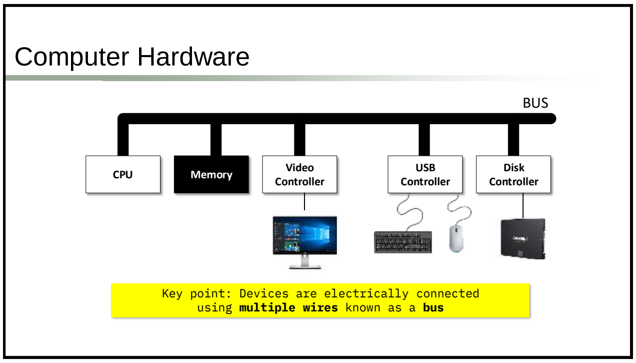
MAIN MEMORY

init executes "runlevel1" program

RUNLEVEL

runlevel programs are executed from /etc/rc.d/rc.*.d/

25



Computer Hardware

Key point: Devices are electrically connected using **multiple wires** known as a **bus**

26

Any Questions?

27