CPSC 457

PRINCIPLES OF OPERATING SYSTEMS

Winter 2019

Pavol Federl

Contains slides from Mea Wang, Andrew Tanenbaum and Herbert Bos

CPSC 457

- From the calendar:

  "An introduction to operating systems principles. Performance measurement; concurrent programs; the management of information, memory and processor resources."

- Goals:
  - understand fundamental operating system concepts:
    - processes, threads, scheduling, memory, I/O, file systems, etc
  - apply data structures and algorithms learned in system development
 - make you a better (system) programmer

What is an Operating System?

Applications

↓ ↑

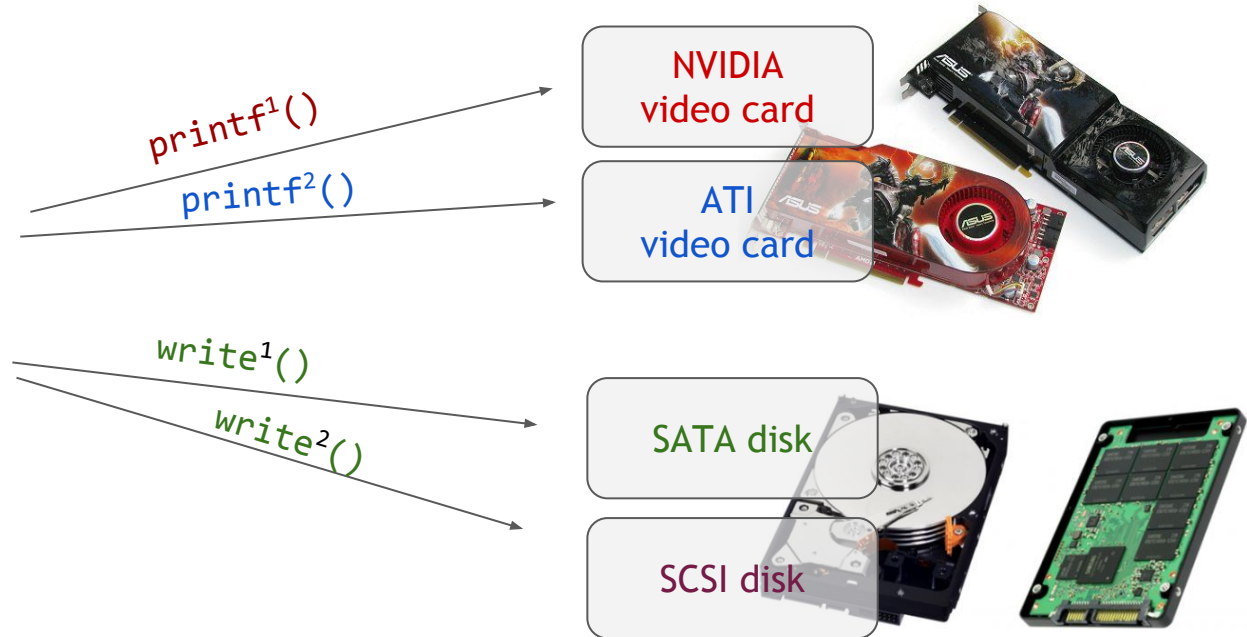| Operating system |
| :---: |

↓ ↑
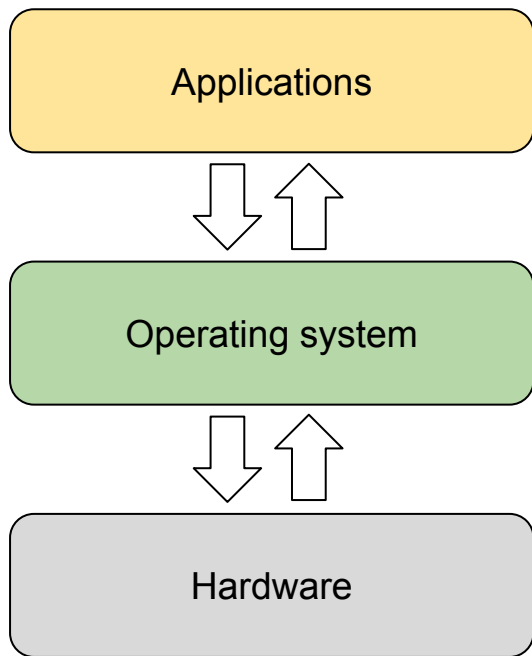
Hardware

- OS is essentially just another software
- OS sits between applications and hardware
- OS deals with many familiar issues:
  - performance: CPU, memory, …
  - resource utilization: scheduling, management, …
  - security: protection, operation mode, …
- understanding OS is key to **system programming**
- useful techniques: data structures, conflict resolution, concurrency, resource management, communication …
- OSes make it 'easier' to write applications
  - why…?

Life without OS

application:

hardware:

NVIDIA
video card
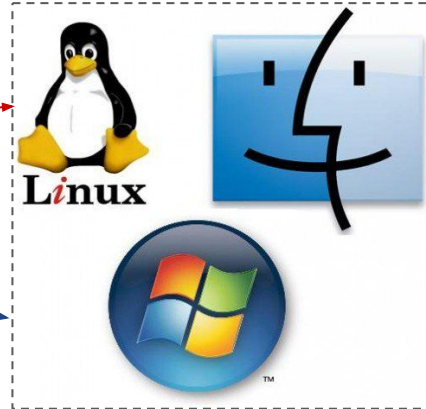
ATI
video card

$printf^1()$

$printf^2()$

$write^1()$
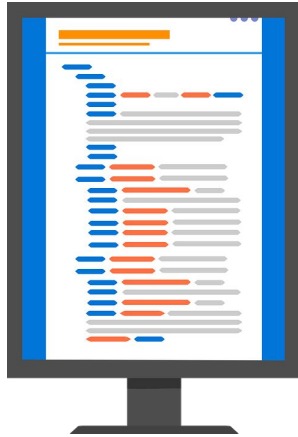
$write^2()$

SATA disk

SCSI disk

Life with OS

application:

operating system:

hardware:

printf()

write()

NVIDIA
video card

ATI
video card

SATA disk

SCSI disk

SSD disk

NFS

What is an OS?

- provides common functions of controlling and allocating resources for application programs

    ○ application programs use hardware

    ○ but hardware is notoriously difficult to use (at low level)

    ○ OS provides higher level abstraction to hardware through APIs

- provides controlled allocation for **efficient** and **fair** resource use

⇒ hides the complexity of the underlying hardware and gives the user /applications

a nicer view of the computer (through APIs, called **system calls**)

Operating systems

- we use different OSes everyday

- Personal Computer (PC) OS: support single user on various tasks (e.g., Windows, Linux, Mac OS)
- Mobile OS: begin to resemble PC OSes (e.g., iPhone, Samsung Galaxy)
- Server OS: multi-users, share hardware/software (e.g., Solaris, FreeBSD, Linux, Mac OS Server and Windows Server)
- Mainframe OS: batch, transaction processing, and time sharing services
- Embedded OS: no untrusted software support (e.g., TV, cars, cell phone, MP3 players)
- Smart Card OS: very primitive, proprietary systems (e.g., on credit cards)
- Sensor Node OS: event driven, special-purpose (e.g., detect forest fire, measure temperature)
- Real-Time OS: deadline driven (e.g., assembly line, multimedia systems)

Logistics

- office hours:
  - ICT 742, Fri 12am - 2pm
- email TA:
  - direct all your questions to your TA
  - get your TAs email in first tutorial
  - subject: **CPSC457** - Assignment question…
- email instructor:
  - ask only questions that your TA was unable to answer
  - address: pfederl@ucalgary.ca
  - subject: **CPSC457** - Assignment extension…
  - include your tutorial # and the name of your TA !!!
- email test …

Websites

- D2L
  - assignment submissions
  - posting grades
  - discussions (if requested)

- google sites (link on D2L)
  - lecture notes
  - assignment specifications
  - announcements
  - links & files
  - …

# Communication

- two way street

- you communicate too:

  - in class - start and end with a Q&A

  - via TAs

  - office hours

  - email

Tutorials

- are NOT optional

- reviews of lecture material

- occasionally new material

  - midterm & final exam will include questions from tutorials

- assignment discussions

- exam reviews/feedback

- start next week?

Textbook
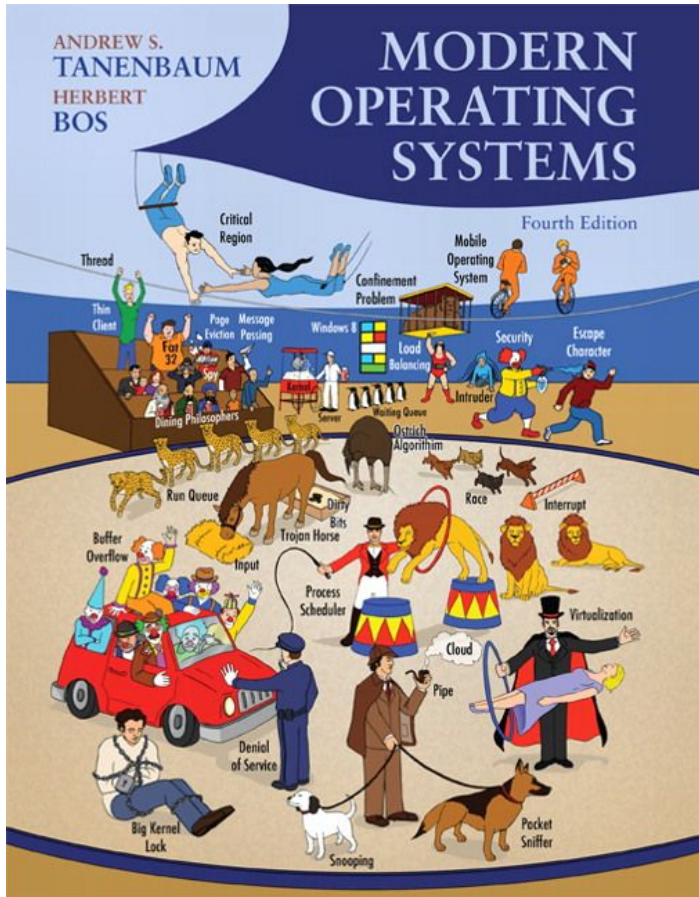
- required *

- should be available in bookstore

- we will cover mostly first 6 chapters

Textbook
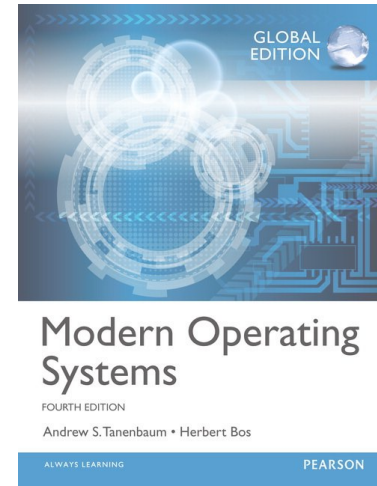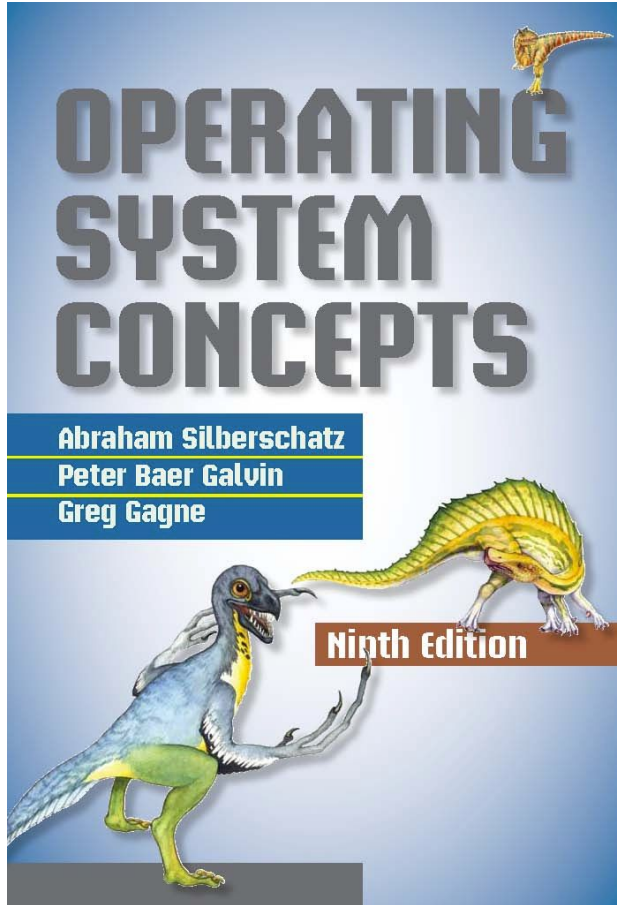
- recommended

- [8th edition available online](#)
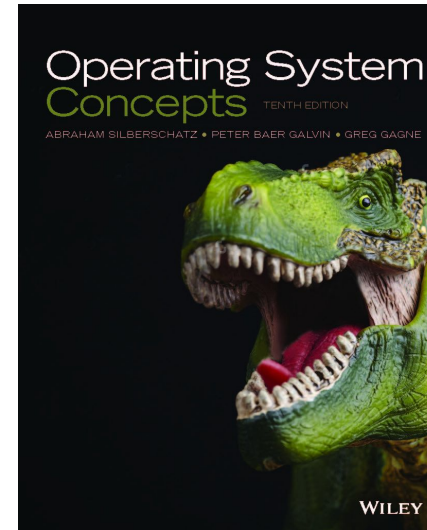
Resources

- lecture slides:
  - will be posted on the course website
  - check website regularly
- slides by themselves are not enough, they are just outlines...
  - take notes!
  - ask questions!
  - take more notes!
- if you miss a lecture, copy someone else's notes

Resources

online resources, e.g.:

http://proquest.safaribooksonline.com.ezproxy.lib.ucalgary.ca/?uicode=ucalgary

check links section on course website

Evaluation

- 5 assignments (30%)

- in-class midterm exam (30%)

  - tentative date: Thursday, Feb 28

- registrar scheduled final exam (40%)

  - date: tbd

- final mark determined from the weighted sum

- **please note**: you need to get C- or better on weighted midterm + final to get C- or better

Assignments

- marked by your TAs – direct your questions to your TAs
- submit electronically via D2L (http://d2l.ucalgary.ca)
  - more info in tutorials
  - may have to ZIP or TAR files before uploading
  - check your submission by re-downloading
- you can work on it during labs, but mostly on your own time
- will require substantial amount of coding
- deadlines are strict - late policy is 100% penalty
- all assignments to be implemented in C or C++ unless explicitly stated otherwise

- **your assignments MUST run on the lab Linux machines!!!**

Assignments

- avoid being late

- start working on assignments early

- submit early, even if incomplete

- submit often (overwrites old submission)

- double check submission

- ask questions

Assignments

- where can you work on your assignments:
  - computer labs (main floor of MS)
  - Linux machines only
- you could work from home, but you will have minimal support
  - work remotely (using SSH)
  - install Linux
    - single OS
    - dual boot
    - virtual machine (vmware, virtualbox, parallels, …)
    - cygwin
- remember, your assignment MUST run on the Linux lab machines!!!

Midterm & final exams

- closed book (no aids)
- will cover material covered in lectures **AND** tutorials
- midterm exam:
  - ~ 75 minutes
  - will cover first ½ of the course
- final exam:
  - ~ 2 hours
  - will cover everything
- more info in future lectures/labs

Group work

- none, unless explicitly stated otherwise

- beware of **plagiarism**, a serious academic misconduct

- all assignments and exams must be **individual** work

- but… you can discuss assignments, just do not copy or let others copy

"A single offence of cheating, plagiarism, or other academic misconduct, on term work, tests, or final examinations, etc., may lead to disciplinary probation or a student's suspension or expulsion from the faculty by the dean, if it is determined that the offence warrants such action."

Please refer to the University Calendar for more details.

UNIVERSITY OF CALGARY Department of Computer Science

Academic Misconduct

Individual Courses

Grade is on hold

Students must NOT discuss the case with the instructor and TAs

Department of Computer Science

Interviewed by the Assistant Head (Academic Integrity)
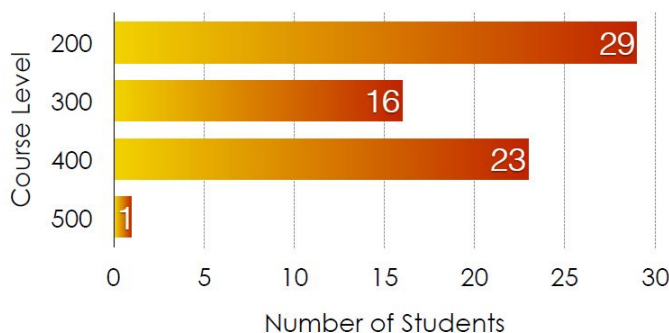
Faculty of Science

Interviewed by the Associate Dean (Undergraduate Programs and Student Affairs)

Academic Penalties

- A failing grade on the work submitted
- A failing grade in the course
- Probation
- Suspension
- Expulsion

Common Offences:

- Copy (with modifications) all or parts of other students' assignments
- Let other students copy your work
- Use online source code without proper citations
- Submit previous year's solution keys as if they are the student's own work

Academic Misconduct in 2017/2018

Course Level

Course Level	Number of Students
200	29
300	16
400	23
500	1

Number of Students (0, 5, 10, 15, 20, 25, 30)

House rules (lectures and tutorials)

- if possible:
    - be on time
    - don't leave early
    - turn off phones
- no recordings without explicit permission

Prerequisites

- CPSC 319 or CPSC 331

- CPSC 325 or CPSC 359 or ENCM 369

- each prerequisite must have been completed with a grade of C- or better

- from U of C Calendar:

"It is the responsibility of students to ensure that they meet all prerequisite requirements. Students who do not meet the exact prerequisites/corequisites and do not have Departmental permission to waive those requirements will have their registrations cancelled automatically by the Faculty after the deadline for student change of registration."

Lectures

- notes you take during lectures are your main resource for the course

- we will cover:

  - Processes

  - Threads

  - Concurrency

  - Scheduling

  - Deadlocks

  - Memory

  - File systems

  - Input/output

  - Multiple Processor Systems

How to succeed

- come to lectures/tutorials and take notes

- read the textbook and take more notes

- ask questions

- do the assignments by yourself

- make good use of the tutorials and the office hours

> *"What I **hear**, I forget.*
> *What I hear and **see**, I remember a little.*
> *What I hear, see and **ask** questions about*
> *or **discuss** with someone else, I begin to understand.*
> *What I hear, see, discuss, and **do**, I acquire knowledge and skill."*
>
> from "Active learning: 101 strategies to teach any subject," Silberman, M. (1996)

Friendly ~~warning~~ advice

Lot of coding (assignments), many new concepts (exams).

Questions?