BIL 301 Operating Systems

Ammar Daskin

Administrivia

BIL 222 vs BIL 301

BIL 222 System Programming

- How userspace apps interact with systems(kernel-space) by mostly using system calls.
- Unix system calls

BIL 301 OS

- Internal structure of OS-kernel-space
- Management of different resources
- And services provided to userspace
- HWs with Linux kernel

Attendance

Weekly in person lectures

- Slides are provided before the lectures
- Some things may not be on the slides or may not be explained fully
    - Take notes in the lecture
- joining the google classroom is mandatory
    - hws, announcements, etc are posted on the classroom!
- Remind me the break-times!

Assignments (Hw)

None of the assignments are graded this semester!

- You do not have to do them to pass this course.

4-5 hw:

- Programming assignments (group projects)
  - kernel compile/build
  - writing kernel module
  - copying data from userspace to kernel-space and vice versa
  - Synchronization
  - security/hacking os
  - Coding standards
    - https://www.kernel.org/doc/html/latest/process/coding-style.html
    - or K&R
- Written problems
  - e.g. scheduling algorithms

Grading

~~30% homework (including programming and written assignments)~~

40% 1-midterm exam

60% 1-final exam

Discussion

~~piazza.com~~

We use only classroom.google.com for this semester.

class code:

- Do not post solutions or any significant part of an assignment.
- Do not post anything not related to the course.
- Ask a question when you would like some help with something
- Post something when you would like to help others with something.

Collaboration and Cheating Policy

- Any kind of plagiarism and cheating are prohibited (Please, refer to the university cheating policy).
- If you benefit from some work of others, list them as references (online references or books)
- Discussing the assignments or projects with your friends is allowed; but, all the submitted work should be yours alone. List your collaborators (if you discuss your homework with your friends) in your assignments.

Textbooks and Course Material

- Operating System Concepts, 10th Edition Abraham Silberschatz, Greg Gagne, Peter B. Galvin, https://www.wiley.com/en-us/Operating+System+Concepts%2C+10th+Edition-p-9781119320913
- **Lectures slides are based on the slides**
  - [https://www.scs.stanford.edu/24wi-cs212/notes/] (https://www.scs.stanford.edu/24wi-cs212/notes/)
  - https://www.os-book.com/OS10/slide-dir/index.html
  - and https://linux-kernel-labs.github.io/
  - weekly posted on classroom.google.com before the lecture.
- **Other resources**
  - Linux Kernel Documentation
  - intel CPU manual
  - OS security and more: https://www.ics.uci.edu/~goodrich/teach/cs201P/notes/
  - ebook for synchronization https://dl.acm.org/doi/book/10.5555/2385452
    - https://booksite.elsevier.com/9780123973375/
  - kernel source code
  - https://linux-kernel-labs.github.io/refs/heads/master/index.html
  - Testing Linux: https://linux-test-project.github.io/

Weekly topics

1. Chapter 1: Introduction
2. Chapter 2: Operating System Structures
3. Chapter 3: Processes
4. Chapter 4: IPC, Threads & Concurrency
5. Chapter 5: CPU Scheduling
6. Chapter 5, 6-7: CPU Scheduling-II, Intro to Synchronization Tools & Examples
7. (not included in the book) Synchronization II (atomic instructions, memory barriers (e.g., mb, fence, volatile), C11 atomic library (relaxed, acquire, release), lock free programming, cache coherency,)
8. Midterm exam
9. Synchronization review, Deadlocks
10. Chapter 9: Main Memory
11. Chapter 10: Virtual Memory
12. Chapter 11-12: I/O Systems
13. Chapter 12-13-14-15: File-System Interface, Implementation, and Internals
14. Protection, Security

Course goals

Introduce you to operating system concepts

- Hard to use a computer without interacting with OS
- Understanding the OS makes you a more effective programmer

Cover important systems concepts in general

- Caching, concurrency, memory management, I/O, protection
- Synchronization and many topics may help you write more efficient user apps

Teach you to deal with(code/compile/build/install) larger software systems

- Programming assignments much larger than many courses
- Compiling linux kernel may take a few hours

Programming assignments

None of the assignments are graded this semester!

- You do not have to do them to pass this course.

Implement/edit parts of linux kernel

- writing new system call
- adding new module
- security

Writing/testing synchronization tools (userspace)

- memory barriers
- thread libraries

First homework (hw0):

- **Download and install linux on virtualbox**
- **Download source code, and compile it…**

Installing Linux on Windows

Windows:

with WSL

https://learn.microsoft.com/en-us/windows/wsl/install

choose any linux distro that are available in appstore

https://learn.microsoft.com/en-us/windows/wsl/use-custom-distro

GUI

https://learn.microsoft.com/en-us/windows/wsl/tutorials/gui-apps?source=recommendations

windows terminal:

https://learn.microsoft.com/en-us/windows/terminal/install

Using virtual machine

If you have linux or windows machine

You can install new OS through virtual machine (prefered method):

VirtualBox, VMware, Parallels

https://www.geeksforgeeks.org/how-to-install-ubuntu-on-virtualbox/

https://ubuntu.com/tutorials/how-to-run-ubuntu-desktop-on-a-virtual-machine-using-virtualbox#1-overview

You can also run linux containers on windows

https://ubuntu.com/tutorials/windows-ubuntu-hyperv-containers#1-overview

macOS

Download linux ARM64 (AArch64) version

https://www.makeuseof.com/how-to-install-virtualbox-apple-silicon-mac/   :

For m3 chips

https://wiki.qemu.org/Main_Page

https://www.qemu.org/download/

https://mac.getutm.app/

Linux test

https://linux-test-project.github.io/