Introduction



CS 351: Systems Programming Michael Lee <lee@iit.edu>



Michael Lee

- lee@iit.edu
- http://moss.cs.iit.edu
- Office hours: Tue/Thu 11AM-1PM
 - By appointment only! (Zoom/In-person)



Agenda

- Syllabus & Administrivia
- Course overview ("Systems Programming")



§Syllabus



Prerequisites

- "substantial" programming experience
- data structures: concepts & implementation
- basic run-time analysis (big O)
- knowledge of (any) assembly language
- computer organization essentials



- computer organization essentials:
 - data representation (binary, two's comp, f.p. inaccuracy, etc.)
 - von Neumann model
 - CPU, memory, I/O
 - stack usage / conventions



Online resources

1.Course website <u>moss.cs.iit.edu/cs351</u>

- static information

moss.cs.iit.edu/cs3	51/		
CS 351: S	ystems Programmin	g	
Announceme	ents		
The Zoom mee	ting ID for our online lectures is 893 7228 5	i322 you can join using <u>thi</u>	s link.
 Lecture record 	ings can be found in the class Panopto fold	er	
please join at <u>h</u>	ttps://discord.gg/exbFK5b		
Set en stern			
Please note that real	dings for a given lecture should ideally be re dings are from "Computer Systems: A Prog		
afterwards. Most rea slides/notes will be u	dings for a given lecture should ideally be re dings are from <u>"Computer Systems: A Prog</u> pdated after class, when available. is tentative and may be updated.		
Please note that real (fterwards. Most real lides/notes will be u The lecture calendar	dings are from <u>"Computer Systems: A Prog</u> pdated after class, when available.		
Please note that real (terwards. Most rea lides/notes will be u "he lecture calendar Dates	dings are from " <u>Computer Systems: A Prog</u> pdated after class, when available. is tentative and may be updated.	rammer's Perspective, 3rd E	idition" (CS:APP3e). Lecture
lease note that real Iterwards. Most real lides/notes will be u he lecture calendar Dates May 25	dings are from " <u>Computer Systems: A Prog</u> pdated after class, when available. is tentative and may be updated. Topic Syllabus and Course overview C Primer	rammer's Perspective. 3rd E Notes	idition" (CS:APP3e). Lecture Reading(s)
Please note that real (fterwards. Most real lides/notes will be u The lecture calendar Dates May 25 May 25 - May 27	dings are from <u>"Computer Systems: A Prog</u> pdated after class, when available. is tentative and may be updated. Toplc Syllabus and Course overview	Notes	idition" (CS:APP3e). Lecture Reading(s) Syllabus
² lease note that real (fterwards. Most real iides/notes will be u The lecture calendar Dates May 25 May 25 - May 27 Jun 01	dings are from " <u>Computer Systems: A Prog</u> pdated after class, when available. is tentative and may be updated. Topic Syllabus and Course overview C Primer	Notes 01-prelim 02-c, x86-64-guide	idition" (CS:APP3e). Lecture Reading(s) Syllabus K&R, <u>Pro Git</u> chapters 1, 2
Please note that real itterwards. Most rea iiides/notes will be u The lecture calendar Dates May 25 May 25 - May 27 Jun 01 Jun 01	dings are from <u>"Computer Systems: A Prog</u> pdated after class, when available. is tentative and may be updated. Topic Syllabus C Primer The Process	nammer's Perspective. 3rd E Notes 01-prelim 02-c. x86-64-guide 03-theprocess	idition" (CS-APP3e). Lecture Reading(s) Syllabus K&R, Pro Git chapters 1, 2 CS:APP 8.1-8.2
Viease note that read Viease note that read lides/notes will be u The lecture calendar Dates May 25 May 25 Jun 01 Jun 01 Jun 03	dings are from <u>"Computer Systems: A Prog</u> pdated after class, when available. is tentative and may be updated. Topic Syllabus and Course overview C Primer The Process Process Management: fork/exit	Notes 01-prelim 02-c, x86-64-guide 03-theprocess 04-procmgmt-1	dition* (CS:APP3e). Lecture Reading(e) Syllabus K&R, Poo Git chapters 1, 2 CS:APP 8.1-8.2 CS:APP 8.3-8.4.2
Viease note that read Viease note that read ulides/notes will be u the lecture calendar Dates May 25 May 25 May 25 Jun 01 Jun 01 Jun 03 Jun 03 Jun 03	dings are from " <u>Connuter</u> Systems: A Prog pdated after class, when available. is tentative and may be updated. Topic Syllabus and Course overview C Primer The Process Process Management: tox/text Process Management: wal/exec	Notes Notes 1-prelim 02-prelim 03-theprocess 04-process 04-process 04-process	Kasding(s) Syllabus K&R, <u>Pro Ott</u> chapters 1, 2 CSAPP 8:18-2 CSAPP 8:38-4.2 CSAPP 8:4-3.8-4.6
Please note that real afterwards. Most rea slides/notes will be u	drigs are from " <u>Conneuter Systems: A Prog</u> pdated after class, when available. is tentative and may be updated. Topic Syllabus and Course overview C Primer The Process Process Management: forklexit Process Management / Unix Shell	Notes 01-prelim 02-cr, x88-64-guide 03-theprocess 04-process 04-process 05-process 05-process 06-process 06-proces 06-process 06-proces 06-proces 06-process 06-proces 06-process 06-process 06-process 06-proces 06-proc	Reading(s) Syllabus K&R, Pro Git chapters 1, 2 CSAPP 8.1-8.2 CSAPP 8.3-8.4.2 CSAPP 8.3-8.4.6 CSAPP 8.5-8.6

- lecture calendar, assignment writeups, slides, screencasts, links, etc.



Online resources

2.Blackboard

- grade spreadsheet
- online exams

🥡 illinois institu	JTE OF TECHNOLOGY	📃 Michael Saelee 👤 🛉 🛛	Log
		My Institution Cours	ses
ft 🖸 Home Page		Edit Mode is:	
+ = = = 6 11	Home Page ©		
 XLSD2201320 (CS-351- Parent.135) 			
Home Page	Add Course Module	Customize Page	†1
Syllabus 📰	My Announcements	My Calendar	
Content 🗉	No Course Announcements have been posted in the last 7 days.	No calendar events have been posted for the next 7 days.	
Discussions Groups	more announcements $ ightarrow$	more calendar events →	
Tools	My Tasks	To Do	
Library Resources	My Tasks:	Edit Notification Settings	
Help	No tasks due.	What's Past Due Actions ¥	
COURSE MANAGEMENT	more tasks \rightarrow	► All Items (0)	
Control Panel		What's Due Actions 😝	
▶ Files →	What's New	Select Date: 01/14/2013 Go	
Course Tools Announcements Blogs	Edit Notification Settings Actions ¥	▼ Today (0) © Nothing Due Today	



Online resources

3.Vimeo channel: screencasts

- <u>vimeo.com/channels/cs351</u>
- walkthroughs & tutorials (check before starting labs!)





Textbooks









Grading

- 50% Labs
- 25% Midterm exam
- 25% Final exam



Grade Scale

```
char letter_grade(float score) {
    if (score >= 90.0) return 'A';
    else if (score >= 80.0) return 'B';
    else if (score >= 70.0) return 'C';
    else if (score >= 60.0) return 'D';
    else return 'E';
```





Labs

- 5-7 fairly substantial machine problems
- real-world application of concepts covered in lecture & textbook
- late policy: elastic over summer
 - hard due date for everything: June 24



Exams

- Midterm exam on June 10
- Final exam is nominally cumulative
- Scores may be linearly scaled so that median/mean (whichever lower) is 70%



§Course Overview



"Systems Programming"

system |'sistəm|

noun

1 a set of connected things or parts forming a complex whole

(New Oxford American Dictionary)



"Systems Programming"

- Programming the operating system
- What does *that* mean?



OS vs. OS kernel

- OS kernel ≈ smallest subset of OS code needed to bootstrap system and provide basic services to user programs
 - "smallest" is debatable



How to "program" it?

- Require some API
 - Application Programming Interface
 - A collection of (documented) functions
 - e.g., get/put/del for a hashtable



Computer Science

OS API

- a.k.a. "system call" interface
 - OS as a very low-level library
- common purpose: provide services to user level programs
 - def: program in execution = process



The Process

- A program in execution
- Code + Data { global, local, dynamic }
 + OS kernel data
- OS hides complexity of machine from processes by creating *abstractions*



AN X64 PROCESSOR IS SCREAMING ALONG AT BILLIONS OF CYCLES PER SECOND TO RUN THE XNU KERNEL, WHICH IS FRANTICALLY WORKING THROUGH ALL THE POSIX-SPECIFIED ABSTRACTION TO CREATE THE DARWIN SYSTEM UNDERLYING OS X, WHICH IN TURN IS STRAINING ITSELF TO RUN FIREFOX AND IT'S GECKO RENDERER, WHICH CREATES A FLASH OBJECT WHICH RENDERS DOZENS OF VIDEO FRAMES EVERY SECOND

> BECAUSE I WANTED TO SEE A CAT JUMP INTO A BOX AND FALL OVER.

"Abstraction"

http://xkcd.com/







Primary Abstractions

- Logical control flow
- Exceptional (extra-process) control flow
- Logical address space
- "I/O" (via uniform APIs)
- Interprocess Communication



In the old days ...

- ... every program had to include its own implementation of all the above!
- Now, OS simplifies life for all of us.
 - Only need to know how to *use* them, not how they're *implemented*.



But!

- In this class we dig a bit deeper
 - What facilities are encapsulated by syscalls?
 - What limitations/restrictions do they have?
 - Why are they designed the way they are?
 - How do they work behind the scenes?



But why should I care?



- *efficiency*: know how to use tools optimally; reuse existing features and design/layer new ones appropriately
- *robustness*: avoid bugs/failures & know how to diagnose and fix them



the real reason: it's fun to take things apart!







goal: turn you into a **hacker**



(or: make you a **better** hacker)



hacker |'hakər|

noun

1 A person who enjoys exploring the details of programmable systems and how to stretch their capabilities, as opposed to most users, who prefer to learn only the minimum necessary.

The Jargon File, version 4.4.7



Our tools (& approach)

- C & Linux
 - C: low-level language
 - GNU Linux: open source kernel & tools
 - GNU gdb & gcc; debugger & compiler



Fourier

- All labs must be tested and submitted on the class Linux server: <u>fourier.cs.iit.edu</u>
 - You will receive an e-mail with account info
 - If off-campus, must connect via IIT VPN
 - Log in via SSH client



Git & GitHub

- All labs are distributed using Git via GitHub
 - Distributed VCS + platform
- Typical workflow: (1) accept invitation and get a private copy of the assignment repo, (2) clone repo on fourier, (3) work on assignment, (4) submit via commit & push

