Introduction



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



Michael (Sae) Lee

- lee@iit.edu
- http://moss.cs.iit.edu
- Office: SB 226C



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



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

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

• • • • • •	S	4 b [0] ;
CS 351: Systems F	rogramming	
Announcements		
 Neizane to the fell 2005 of the 	er of CS 351: Systems Programming	
Calendar	Machine Problems	Resources
August 24	Plane relat	Ad win issriela
 State Syllecture & Overview August 28 - Sequencher 2 State C State C, Informat C, Danne C, Martin B, C State C, Primer C, Danne C, Barre D, C Fanning CM Fanning CM Fanning CM Fanning CM Fanning CM Fanning CM State C, The Presence State C (the Presence SCO) Fanning CCM PLLAS 	Start work/ foot with arbitra M and/proof - i.e. the sectors with arbitra indexing event through the entire write up before an fing to ende event through the entire write up the sectors are needed. is a sector arbitrary arbitrary is needed an ignments. He planker will read acceptes included getting a zero. Chellenbalan (k public) v The Sky and despace Lamb.	 Instruction Michael Society Social International Control of Contro of Control of Control of Control of Contro of Control of Cont
September 3 - 14	 Assignanti TBAI Inas: TIM 	 Counterry links Class of accuration for the on Plasma Versus discussed
 Tapin The LBAX schools management All Wider Process Mechanism ed. 1 	 <u>Calify a Humanian</u> (of pairs) Analyzani TBA; 	Primary reference materials



2.Blackboard

- only for grade reporting!

🕅 ILLINOIS INSTIT	UTE OF TECHNOLOGY	Total and T 4 6
		Ny Institution Dears
11 D Feasible		bit Mule is
	Home Page ©	
STORE 100 (19-150) 👘		
tone face	Add Cauror Webde	Enderiar Payr
1464 B	My Association	wy Colemian
dermake p	No face we descent sector was been been possed in the law (Mandradar reach incredient partial in the and 2 date.
A READ	skeps.	
inage.	more announcements and	and introduction of the
141	Mr. Teatra	TaDa
bary income	he to the total of total of the total of tot	10 Kollostochellings
nto	No costa due.	WHAT'S Pull Pole Address y
CODE NAME ON LN -	enertada	k dikensiti 📀
in all family		What's Due Atlant v
161	wher's new	Select Dette: an year work 📓 Det
Dutik Tek a	Methodor Scinci Actors y	■ Stolay ⊕ ⊕ Uniting Data Teslay



3.Vimeo channel: screencasts

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





4. Piazza: discussion forum

- all class-related questions
- monitored by TAs



- announcements, links to additional readings & resources





Textbooks









Grading

- 50% Labs
- 25% Midterm exam
- 25% Final exam
 - exam scores *normalized* to 70%
 - need $\geq 50\%$ on both exams to pass



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

- fairly substantial machine problems
 - 100-1000 LOC
- real-world application of concepts covered in lecture & textbook
- 1-3 weeks allotted for each



§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
- Uniform I/O
- 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

