

```
$ cat welcome.c
#include <stdio.h>

int main(int argc, char *argv[])
{
    printf("Welcome to COS 217\n");
    printf("Introduction to Programming Systems\n\n");
    printf("%s %d\n", "Spring", 2021);
    return 0;
}
```

```
$ cat Makefile
CC=gcc217
welcome: welcome.o
```

```
$ make
gcc217      -c -o welcome.o welcome.c
gcc217     welcome.o      -o welcome
```

```
$ ./welcome
```

```
Welcome to COS 217
Introduction to Programming Systems
```

```
Spring 2021
```

Agenda



Course overview

- **Introductions**
- Course goals
- Resources
- Grading
- Policies

Getting started with armlab

- Brief overview of Linux and bash
- bash walkthrough (separate video)

Introductions



Lead Instructor

- Szymon Rusinkiewicz smr@cs.princeton.edu

Lead Preceptors

- Christopher Moretti cmoretti@cs.princeton.edu
- Donna Gabai dgabai@cs.princeton.edu

Preceptors

- Scott Karlin scott@cs.princeton.edu
- Weicong Dong weicongd@princeton.edu
- Juan Duque jduque@princeton.edu
- Anne Kohlbrenner akohlbrenner@princeton.edu
- Huihan Li huihanl@princeton.edu

Agenda



Course overview

- Introductions
- **Course goals**
- Resources
- Grading
- Policies

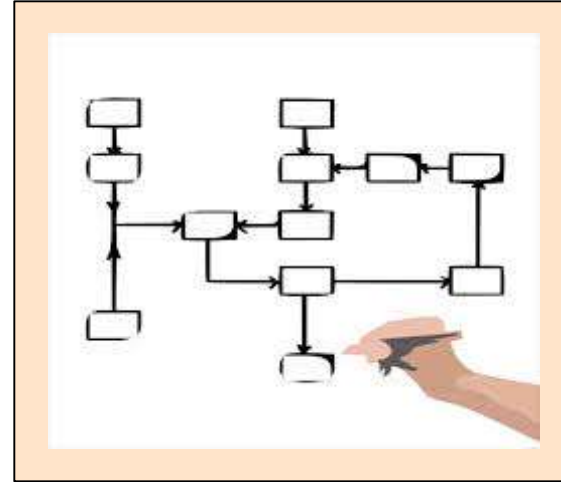
Getting started with armlab

- Brief overview of Linux and bash
- bash walkthrough (separate video)



Goal 1: Programming in the Large

Learn how to compose large(r) computer programs



Topics

- Modularity/abstraction, information hiding, resource management, error handling, testing, debugging, performance improvement, tool support



Goal 2: Lower-level Languages

```

int main(void) {
    while ((iChar = getchar()) != EOF) {
        lCharCount++;
        if (isspace(iChar)) {
            if (iInword) {
                lWordCount++;
                iInword = FALSE;
            }
        }
    }
}

```



```

main:
.LFB0:
.cfi_startproc
stp x29, x30, [sp, -16]!
.cfi_def_cfa_offset 16
.cfi_offset 29, -16
.cfi_offset 30, -8
add x29, sp, 0
.cfi_def_cfa_register 29
b .L2

```

RELOCATION RECORDS FOR [.eh_frame]:

OFFSET	TYPE	VALUE
0000000000000001c	R_AARCH64_PREL32	.text

Contents of section .text:

```

0000 fd7bbfa9 fd030091 39000014
00000090  .{.....9.....

```



Goals: Summary



Help you to gain ...



[Jungwoo Hong](#)

Programming Maturity

Specific Goal: Learn C



Question: Why C instead of Java?

Answer 1: A primary language for “under the hood” programming in real code bases.

Answer 2: A variety of experience helps you “program in the large”





Specific Goal: Learn Linux

Question: Why use the Linux operating system?

Answer 1: Linux is the industry standard for servers, embedded devices, education, and research

Answer 2: Linux (with GNU tools) is good for programming (which helps explain answer 1)

Linux™



Agenda



Course overview

- Introductions
- Course goals
- **Resources**
- Grading
- Policies

Getting started with armlab

- Brief overview of Linux and bash
- bash walkthrough (separate video)

Lectures



Describe material at a mix of levels

- Some conceptual (high) overview
- Some digging into details

Videos on YouTube, Slides on course website



Lectures available by:	Friday	Monday afternoon
Watch lecture before:	Monday morning	Wednesday morning
Lecture Q&A:	M 10:00-11:00	W 10:00-11:00
Precept:	M / T	W / Th

Lecture Q&A



Live Q&A sessions via Zoom

- For material from current lectures
- *Please watch* the lecture first
- Will be recorded to accommodate timezones
- If you can't make it, submit question ahead of time (via email to smr@cs.princeton.edu)



Lectures available by:	Friday	Monday afternoon
Watch lecture before:	Monday morning	Wednesday morning
Lecture Q&A:	M 10:00-11:00	W 10:00-11:00
Precept:	M / T	W / Th



Precepts

Describe material at the “practical” (low) level

- Support your work on assignments
- Handouts available via course website

Etiquette

- Attend your precept: attendance will be taken
 - Must miss your precept? ⇒ inform preceptors & attend another
- Use TigerHub to move to another precept

Precepts begin Monday / Tuesday!

Lectures available by:	Friday	Monday afternoon
Watch lecture before:	Monday morning	Wednesday morning
Lecture Q&A:	M 10:00-11:00	W 10:00-11:00
Precept:	M / T	W / Th

Websites



<https://www.cs.princeton.edu/~cos217> (Course website)

- Home page, schedule page, assignment page, policies page

<https://princeton.instructure.com/courses/2065> (Canvas)

- Links to Zoom precepts, Ed, recorded lectures and precepts, Library reserves and other readings, NameCoach



Ed



<https://us.edstem.org/us/courses/4087/discussion/>

- Also available as a Canvas link
- Instructions provided in first precept



Etiquette

- Study provided material before posting question
 - Lecture slides, precept handouts, required readings
- Read / search all (recent) Ed threads before posting question
- Don't reveal your code!
 - See course policies



Books

C Programming: A Modern Approach (Second Edition) (required)

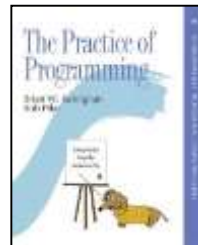
- King
- C programming language and standard libraries

ARM 64-bit Assembly Language (online)

- Pyeatt with Ughetta

The Practice of Programming (online)

- Kernighan & Pike
- “Programming in the large”



Manuals



Manuals (for reference only, available online)

- *ARMv8 Instruction Set Overview*
- *ARM Architecture Reference Manual*
- *Using `as`, the GNU Assembler*

See also

- Linux *man* command



Help!



Office Hours (starting Monday 2/1)

- Preceptors: 2 hours scheduled every weekday + Sunday
- Me: sign up for appointment via <https://calendly.com/smr-princeton>
- Schedule is on the course website
- Links are on Ed

Lab TAs

- Your peers are available 4 hours per day, every single day
- These are specific to debugging your assignments.
For conceptual help with course materials, go to office hours.
- <https://labta.cs.princeton.edu/>

Agenda



Course overview

- Introductions
- Course goals
- Resources
- **Grading**
- Policies

Getting started with armlab

- Brief overview of Linux and bash
- bash walkthrough (separate video)

Grading



Course Component	Percentage of Grade
Assignments *	66
Midterm Exam **	10
Final Exam **	20
Participation ***	4

* 6 assignments × 11% each; penalties for lateness

** During midterms week and final exam period, respectively

*** Did your involvement benefit the course?

- As measured through precept attendance, precept participation, and Ed participation
- Scaled down from prior terms due to being online



Programming Assignments

Regular (every 1.5-2.5 weeks) assignments

0. Introductory survey
1. “De-comment” program
2. String module
3. Symbol table module
4. Directory and file trees *
5. Assembly language programs *
6. Buffer overrun attack *

*(partnered assignment)



Assignments 0 and 1 are available now.

Start early!!

Agenda



Course overview

- Introductions
- Course goals
- Resources
- Grading
- Policies

Getting started with armlab

- Brief overview of Linux and bash
- bash walkthrough (separate video)

Policies



Learning is a collaborative activity!

- Discussions with others that help you understand concepts from class are encouraged

But programming assignments are graded!

- Everything that gets submitted for a grade must be exclusively your own work
- Don't look at code from someone else, the web, Github, etc. – see the course “Policies” web page
- Don't reveal your code or design decisions to anyone except course staff – see the course “Policies” web page



Violations of course policies

- Typical course-level penalty is 0
- Typical University-level penalty is suspension

Sanity



COS 1xx/2xx courses are hard under the best of circumstances

- Information-dense
- Programming is a new skill, or “craft”: not like writing essays or doing problem sets

These are not the best of circumstances

- We are all worried about ourselves, friends, family
- We all feel stressed, anxious, isolated – but when these veer into panic or depression...

Say something, and get help

- Reach out to CPS, your residential college dean, course staff
- No judgment – the rest of us are feeling it too

Agenda



Course overview

- Introductions
- Course goals
- Resources
- Grading
- Policies

Getting started with armlab

- Brief overview of Linux and bash
- bash walkthrough (separate video)

```
Package: android-sdk 26.1.1-1 (Mon Feb 1  
Installing runtime dependencies...  
Installing buildtime dependencies...  
Installing sources...  
Installing sdk-tools-linux-4333796.zip...  
% Received % Xferd Average Speed Time  
Dload Upload To  
100 147M 0 0 4682k 0 0:00  
android-sdk.sh  
android-sdk.csh  
android-sdk.conf  
license.html  
Installing source files with sha1sums...  
73796.zip ... Passed
```

Terminal window showing system status:

- File Edit View Search Terminal Help
- 1 [
- 2 [|
- 3 [|
- 4 [| |
- Mem [||||||||| 2.0G/2.0G
- Swp [||||| 4G/4G

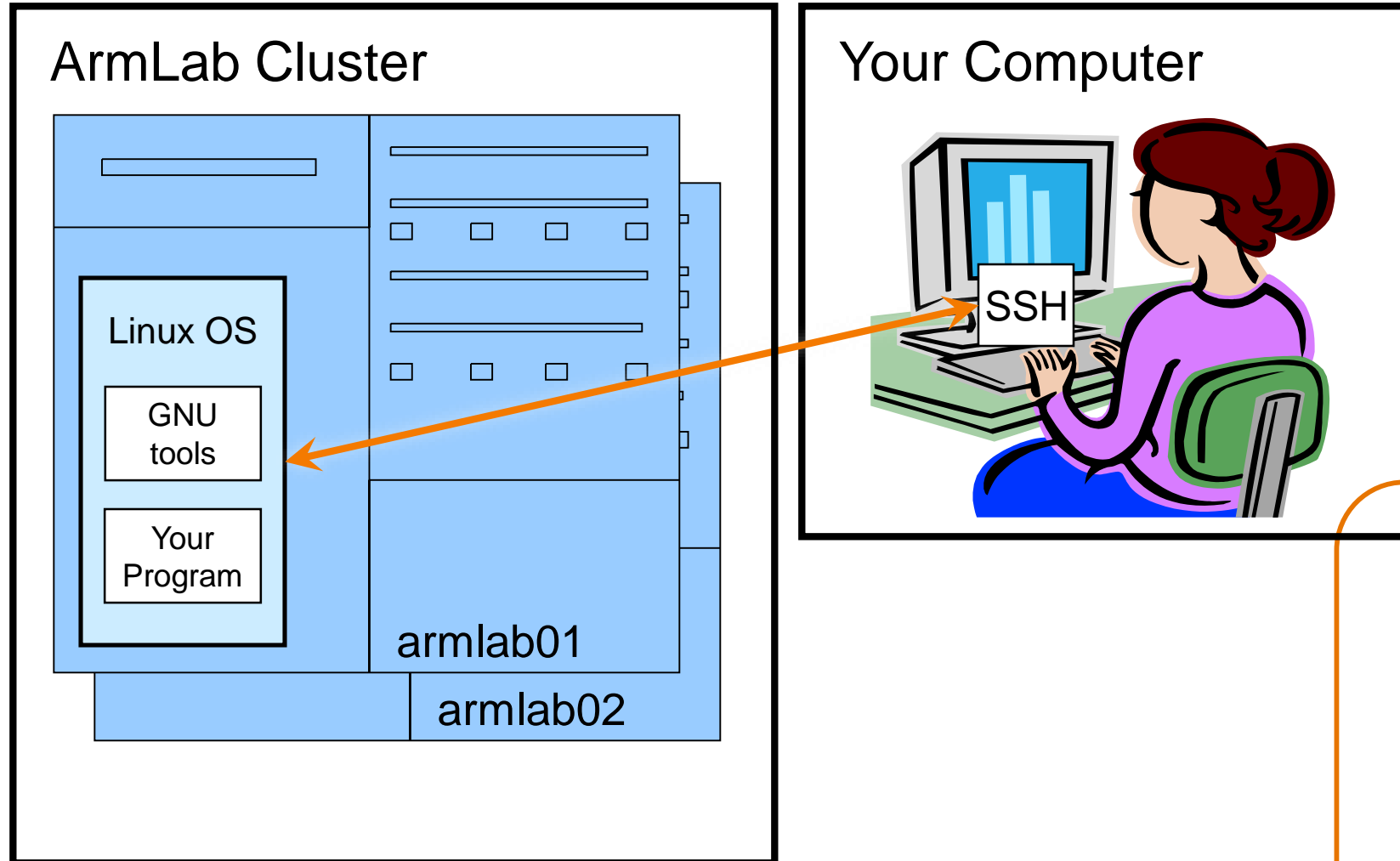
PID	USER	PR	NI	U	St	T	S	...
31208	saikiran	20	0	0	R	0	0	...
22651	saikiran	20	0	0	R	0	0	...
902	saikiran	20	0	0	R	0	0	...
472	root	20	0	0	R	0	0	...
21853	saikiran	20	0	0	R	0	0	...

Programming Environment



Server

Client





Terminology: Terminal vs Shell



[@daniel von appen](#)

[@noxah](#)



Client/Server Implication



You can do this course from anywhere in the world!

- Good in general, when compared with being confined to a cluster in the Friend basement.
- Necessary in these times



Getting Started

Watch the second part of lecture 1

Check out course website **soon**

- **Study “Policies” page**
- Assignments 0 and 1 are available

Establish a reasonable computing environment **soon**

- Instructions given in first precept
- Whatever you choose, you’ll need to get up to speed on Linux at least a little bit, so that will be the second part of this lecture.

Questions?



See you all at the Q&A on Monday!