

CIS 330

C++ and Unix

Lecture 1

Unix

Course Website & Info

- jeewhanchoi.com/uocis330s22/
 - Will be available later today
- Description
 - Practical software design and programming activities in a **C/C++** and **Unix** environment, with emphasis on the details of C/C++ and good programming style/practices.
- Prerequisite
 - CIS 314 (Computer Organization)

Instructor & TA

- Instructor
 - Jeewhan Choi (Jee)
- TA
 - Tamanna Saini
- Office Hours
 - Instructor
 - MW 11:30 – 12:30 *and by appointment* @ DES 328
 - TA
 - M 12:30 – 1:30, T 1:00 - 3:00, and Thursday 12:00 - 1:00 @ DES 237

Textbook

- Physical textbook is not required for this class
- See the class website for online text and resources

Grading

Criteria	Percentage
Homework	40%
Lab	10%
Quiz	10%
Midterm	20%
Final	20%

Grading

Score	Letter Grade
97 - 100	A+
93 - 96	A
90 - 92	A-
87 - 89	B+
83 - 86	B
80 - 82	B-
77 - 79	C+
73 - 76	C
70 - 72	C-
67 - 69	D+
63 - 66	D:
60 - 62	D-
< 60	F

Scores will NOT be rounded up. For example, 96.9 is a A.

Homework

- Assigned every Friday
 - Due the following Friday, 11:59 PM PST (usually)
- Lab attendance is required - grade will be based on lab submission & attendance
 - One lowest lab grade will be dropped
- Quiz will be given randomly on the materials covered in the previous class
 - Solutions will be discussed during class
- Submission of any homework/lab/quiz will **NOT** be allowed after it has been discussed in class

Grading

- All homework will be graded on **functionality** and **aesthetics**
 - Proper use of **comments**, white space, **indentation**, intuitive variable names, etc.
- **Code that does not compile** will be given 0
 - Must compile with “-std=c11” for C code, and “-std=c++14” for C++ code
 - Must compile and run on ix-dev (with the software available for everyone on the system)
- **Late homework will not be graded**, except
 - prior arrangement have been made *at least* 24 hours prior to the due date, or
 - **documented** emergencies
- Use version control
 - e.g., one single large commit may be subject to point deduction
- Develop code in Unix/Linux environment
 - e.g., any sign of Visual Basic or non-Unix/Linux environment may be subject to point deduction

Exams

Midterm

Wednesday, Apr. 20th (week 4)

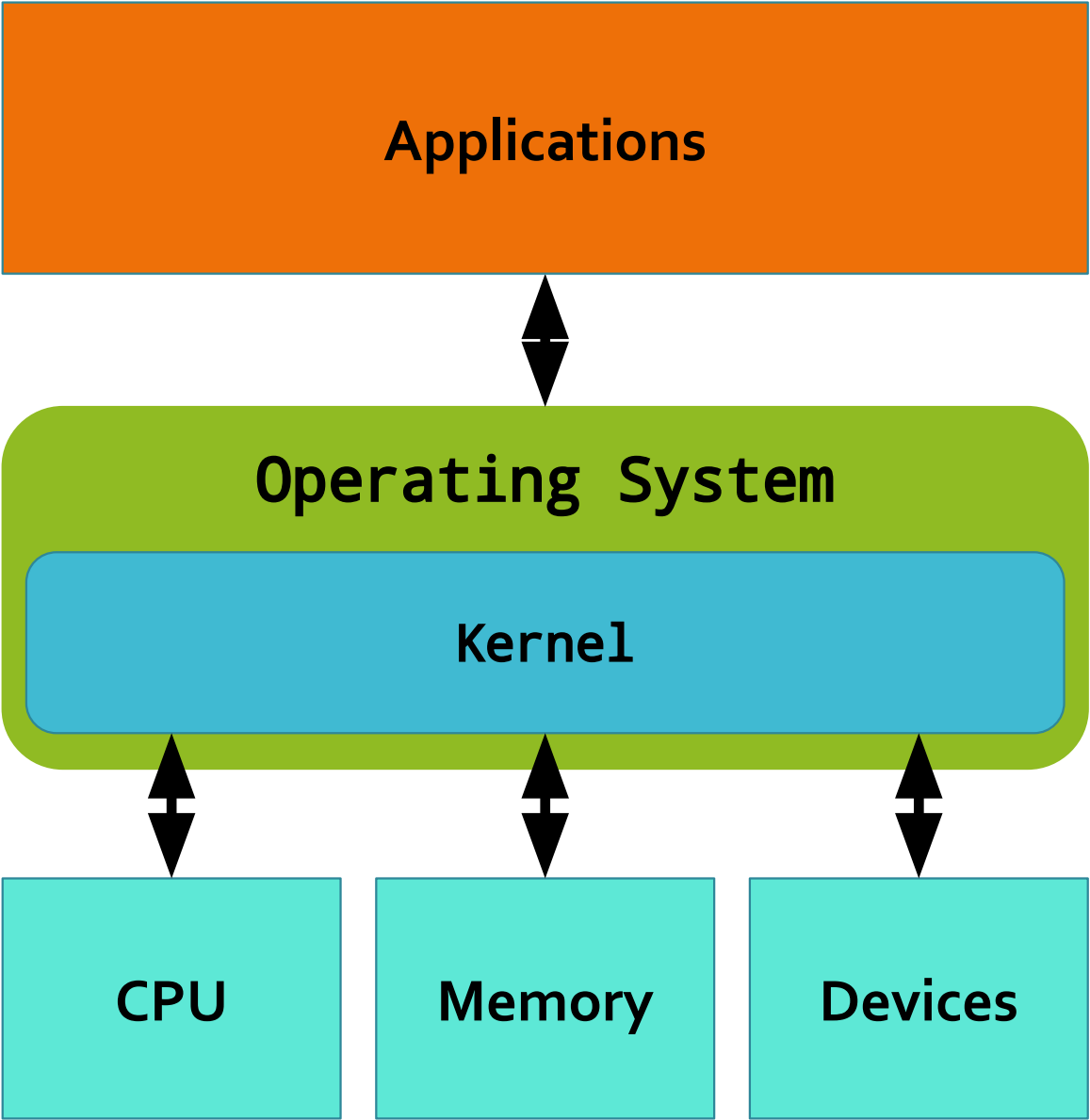
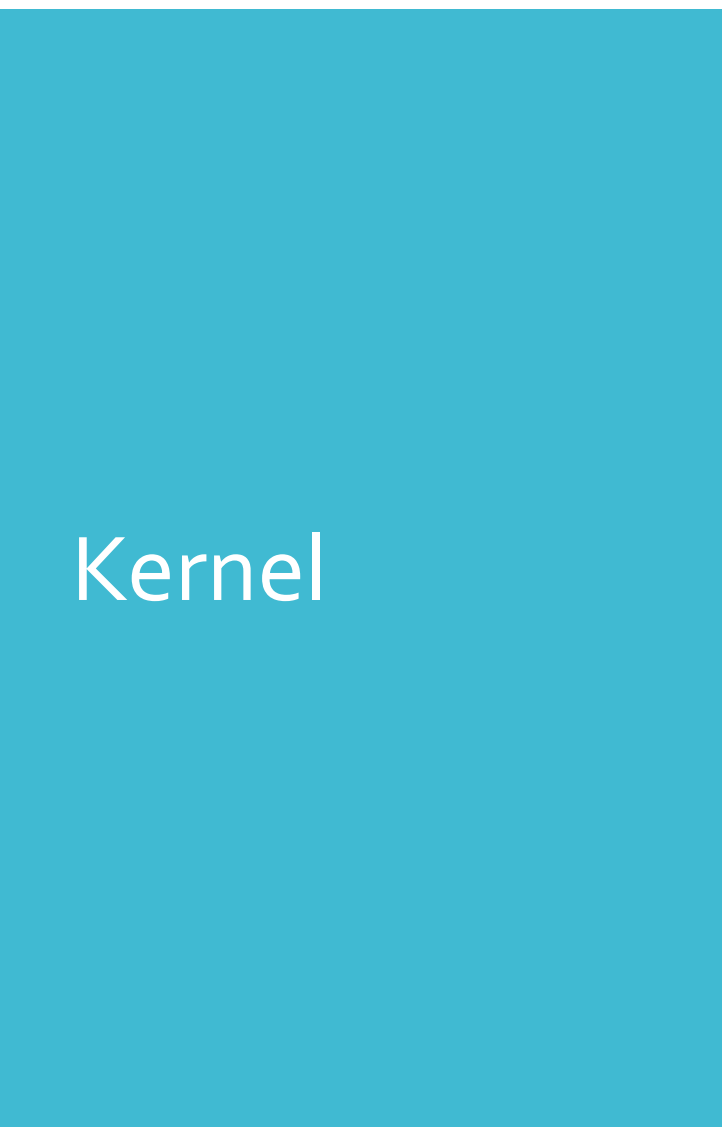
Final

Friday Jun. 10 10:15 - 12:15 (2 hours)



Questions?





Unix

- Class of operating systems (OS) derived from AT&T Unix from Bell Labs
 - Berkeley Software Distribution (BSD), IBM AIX, etc.
- Modular design – OS provides a set of simple tools that each performs a specific task
- Portable – almost entirely written in C
- Portable operating systems Interface (POSIX)
 - MacOS is POSIX compliant

Unix vs. Linux

- Linux kernel was initially developed by Linus Torvald as a truly **free** and **open source** alternative to Unix
- 251 active distributions (as of 1/1/2021, distrowatch.com)
- Two are similar (e.g., most Linux distributions are mostly Portable Operating System Interface (POSIX) compliant)

How do I get access?

- Sign up for an account <https://systems.cs.uoregon.edu/>
- Sign in to ix-dev.cs.uoregon.edu (Ubuntu Linux) using your favorite ssh client
 - e.g., `ssh <username>@ix-dev.cs.uoregon.edu`

User interface

- Terminal window (e.g., terminal in OS X, putty in Windows)
 - Emulates the original 80 x 24 window found on earlier UNIX systems
- Command line interpreter
 - bash
 - Reads the command typed by user and executes those commands

Things you can try

```
jeec@ix-dev: ~ 6$ uname -a
```

```
Linux ix-dev 4.18.0-13-generic #14-Ubuntu SMP Wed Dec 5  
09:04:24 UTC 2018 x86_64 x86_64 x86_64 GNU/Linux
```

```
jeec@ix-dev: ~ 2$ pwd
```

```
/home/users/jeec
```

```
jeec@ix-dev: ~ 3$ date
```

```
Tue 17 Dec 2019 01:17:50 PM PST
```

```
jeec@ix-dev: ~ 4$ date;pwd
```

```
Tue 17 Dec 2019 01:18:29 PM PST
```

```
/home/users/jeec
```

Things you can try

```
jeec@ix-dev: ~ 5$ echo this  
this
```

```
jeec@ix-dev: ~ 6$ echo "this"  
this
```

```
jeec@ix-dev: ~ 7$ echo 'this'  
This
```

```
jeec@ix-dev: ~ 7$ echo Hello World  
Hello World
```

Things you can try

```
jeec@ix-dev: ~ 8$ echo A semi-colon ; is a command separator
```

What would this print on the screen?

Things you can try

```
jeec@ix-dev: ~ 8$ echo A semi-colon ; is a command separator  
A semi-colon  
is: command not found
```

```
jeec@ix-dev: ~ 11$ echo A semi-color ';' is a command  
separator
```

```
A semi-color ; is a command separator
```

Things you can try

```
jeec@ix-dev: ~ 6$ uname -a
```

```
Linux ix-dev 4.18.0-13-generic #14-Ubuntu SMP Wed Dec 5  
09:04:24 UTC 2018 x86_64 x86_64 x86_64 GNU/Linux
```

```
jeec@ix-dev: ~ 6$ uname -m
```

```
x86_64
```

```
jeec@ix-dev: ~ 6$ uname -n
```

```
ix-dev
```

```
jeec@ix-dev: ~ 20$ uname -n -m
```

```
ix-dev x86_64
```

```
jeec@ix-dev: ~ 20$ uname -nm
```

```
ix-dev x86_64
```

Things you can try

```
jeec@ix-dev: ~ 6$ uname -a
```

```
Linux ix-dev 4.18.0-13-generic #14-Ubuntu SMP Wed Dec 5  
09:04:24 UTC 2018 x86_64 x86_64 x86_64 GNU/Linux
```

```
jeec@ix-dev: ~ 6$ uname --all
```

```
Linux ix-dev 4.18.0-13-generic #14-Ubuntu SMP Wed Dec 5  
09:04:24 UTC 2018 x86_64 x86_64 x86_64 GNU/Linux
```

File system

```
CaDS
├── book.tex
├── ch00
│   ├── Preface.pdf
│   ├── Preface.synctex.gz
│   └── Preface.tex
├── ch01
│   ├── ch01.pdf
│   ├── ch01.synctex.gz
│   └── ch01.tex
├── ch02
│   ├── adm3a.jpg
│   ├── ch02.pdf
│   ├── ch02.synctex.gz
│   └── ch02.tex
├── me
│   ├── 20170706-shopping.list
│   ├── book
│   ├── calendar.data
│   ├── Music
│   ├── Pictures
│   ├── shopping.list
│   └── src
├── me.tgz
├── model33teletype.jpg
├── Outline.pdf
├── Outline.synctex.gz
└── Outline.tex
```


Linux file system structure

- /bin - contains basic programs (like bash) that are required during the boot process;
- /sbin - contains programs that must be accessed by the system administrator;
- /lib (or /lib32 or /lib64)- contains dynamic libraries and static support files needed in the boot process;
- /etc - contains configuration files for the system;
- /home - contains the home director for each user

List files

```
jeec@ix-dev: ~ 17$ ls  
etc/ Library/ public_html/ WELCOME.txt
```

Permissions

- Unix/Linux is a multi-user OS

```
jeec@ix-dev: ~ 73$ ls -la jee.txt
```

```
-rw-r--r-- 1 jeec jeec 238 Oct 15 09:09 jee.txt
```

d rwx rwx rwx

Read, write, and execute permissions for **all others**

Read, write, and execute permissions for **group**

Read, write, and execute permissions for **owner**

- indicates a **regular file**
d indicates a **directory**

chmod

- Changes file permissions
 - $r - 4$ (100)
 - $w - 2$ (010)
 - $x - 1$ (001)

```
jeec@ix-dev: ~ 79$ ls jee.txt -la
-rw-rw-r-- 1 jeec jeec 103 Mar 11 17:39 jee.txt
```

```
jeec@ix-dev: ~ 80$ chmod 777 jee.txt
jeec@ix-dev: ~ 81$ ls jee.txt -la
-rwxrwxrwx 1 jeec jeec 103 Mar 11 17:39 jee.txt*
```

```
jeec@ix-dev: ~ 83$ chmod 741 jee.txt
jeec@ix-dev: ~ 84$ ls jee.txt -la
-rwxr----x 1 jeec jeec 103 Mar 11 17:39 jee.txt*
```

Change directory

```
jeec@ix-dev: ~ 27$ cd public_html/  
jeec@ix-dev: ~/public_html 28$
```

cat
(concatenate)
– to print
content of
file(s)

```
jeec@ix-dev: ~ 3$ cat WELCOME.txt
```

Welcome to your new account!

Information about our network or installed software is available on
<<http://systems.cs.uoregon.edu/>>.

You can mail questions about the systems, network or software to
<systems@cs.uoregon.edu>.

Good luck!

cat
(concatenate)
– to create a
new file

```
jeec@ix-dev: ~ 10$ cat > myname
```

```
Jee Whan Choi
```

```
jeec@ix-dev: ~ 11$ cat myname
```

```
Jee Whan Choi
```

Output redirection

```
jeec@ix-dev: ~ 80$ ls -la jee.txt > test1
jeec@ix-dev: ~ 81$ cat test1
-rw-rw-r-- 1 jeec jeec 89 Dec 17 13:49 jee.txt
```

```
jeec@ix-dev: ~ 89$ argh
```

Command 'argh' not found, did you mean:

command 'arch' from deb coreutils (8.30-1ubuntu1)

command 'args' from deb ucommon-utils (7.0.0-16ubuntu1)

Try: apt install <deb name>

```
jeec@ix-dev: ~ 90$ argh 2> test2
```

Also, **>&** and **2>&1** (redirect standard output 2 (stderr) to 1 (stdout)).

touch

```
jeec@ix-dev: ~ 15$ touch test
```

```
jeec@ix-dev: ~ 16$ cat test
```

```
jeec@ix-dev: ~ 17$
```

rm

```
jeec@ix-dev: ~ 18$ rm test
```

```
rm: remove regular empty file 'test'? y
```

OPTIONS

Remove (unlink) the FILE(s).

-f, --force

ignore nonexistent files and arguments, never prompt

-r, -R, --recursive

remove directories and their contents recursively

rm

- Deleted files are gone for good (i.e., there are no trash bins)
- It is extremely difficult to recover deleted files (so be very careful)
- One way to make sure you don't accidentally delete files is to alias the rm command to another one.
 - alias rm='rm -i' -> this prompts the user to confirm deletion (default on ix-dev)
 - alias rm='mv -t ~/trash' -> this moves the file to the trash directory instead of deleting it (i.e., you have emulated Windows/MacOS trash can). You can later remove it for good (after unaliasing rm) if you are sure you don't need it anymore.

mkdir
rmdir

```
jeec@ix-dev: ~ 34$ mkdir new_directory
```

```
jeec@ix-dev: ~ 35$ rmdir new_directory
```

cp
mv

```
jeec@ix-dev: ~ 6$ touch my_file
```

```
jeec@ix-dev: ~ 7$ mkdir my_directory
```

```
jeec@ix-dev: ~ 8$ cp my_file my_directory/
```

```
jeec@ix-dev: ~ 9$ ls
```

```
etc/ Library/ my_directory/ my_file public_html/ WELCOME.txt
```

```
jeec@ix-dev: ~ 10$ ls my_directory/
```

```
my_file
```

```
jeec@ix-dev: ~ 11$ mv my_file my_directory/
```

```
jeec@ix-dev: ~ 12$ ls
```

```
etc/ Library/ my_directory/ public_html/ WELCOME.txt
```

grep & regular expressions

- Pattern matching

<code>^</code>	Beginning of line
<code>\$</code>	End of line
<code>.</code>	Single character
<code>*</code>	Zero or more occurrences
<code>[xyz...]</code>	One occurrence of any digit or character between []
<code>[x-y]</code>	Digit or character from x to y (range)
<code>\+</code>	One or more occurrences of the previous character
<code>\?</code>	Zero or one occurrence of the previous character

grep examples

jee.txt

```
Jee Whan Choi is an assistant professor.  
Jee Choi teaches CIS 330.  
Jee Choilikes coffee.
```

```
jeec@ix-dev: ~ 114$ grep "Jee Choi" jee.txt  
Jee Choi teaches CIS 330.  
Jee Choilikes coffee.
```

```
jeec@ix-dev: ~ 118$ grep "[0-9]\+" jee.txt  
Jee Choi teaches CIS 330.
```

```
jeec@ix-dev: ~ 119$ grep "[0-9][0-9][0-9][0-9]" jee.txt
```

What will this print?

grep examples

jee.txt

Jee Whan Choi is an assistant professor.

Jee Choi teaches CIS 330.

Jee Choilikes coffee.

```
jeec@ix-dev: ~ 114$ grep "Jee Choi" jee.txt
```

Jee Choi teaches CIS 330.

Jee Choilikes coffee.

```
jeec@ix-dev: ~ 118$ grep "[0-9]\+" jee.txt
```

Jee Choi teaches CIS 330.

```
jeec@ix-dev: ~ 119$ grep "[0-9][0-9][0-9][0-9]" jee.txt
```

```
jeec@ix-dev: ~ 120$
```


grep examples

jee.txt

Jee Whan Choi is an assistant professor.

Jee Choi teaches CIS 330.

Jee Choilikes coffee.

```
jeec@ix-dev: ~ 114$ grep -v teaches jee.txt
```

Jee Whan Choi is an assistant professor.

Jee Choilikes coffee.

pipng

- Form of “re-direction,” where output of one command is used as input for another

```
jeec@ix-dev: ~ 13$ cat my_file.txt | head -7 | tail -5
```

- cat my_file.txt -> take the first 7 lines (head) -> take the last 5 lines (tail) of the first 7 lines (i.e., lines 3 through 7)

```
jeec@ix-dev: ~ 52$ cat jee.txt
```

Jee Whan Choi is an assistant professor

Jee Choi is teaching CIS 330

Jee Choi likes coffee

```
jeec@ix-dev: ~ 53$ cat jee.txt | grep "Jee Choi" | wc -l
```

2

- 2 lines contain the phrase “Jee Choi.” (wc -l counts the number of lines)

WC

- Word count (wc)

```
jeec@ix-dev: ~/src/mttkrp_bench 38$ wc common.c
```

```
63  164 1217 common.c
```

- Lines, words, characters

```
jeec@ix-dev: ~/src/mttkrp_bench 44$ wc -l common.c
```

```
63 common.c
```

```
jeec@ix-dev: ~/src/mttkrp_bench 45$ wc -w common.c
```

```
164 common.c
```

```
jeec@ix-dev: ~/src/mttkrp_bench 46$ wc -c common.c
```

```
1217 common.c
```

vim

- Text editor
- Modes
 - Command – any key typed corresponds to a vim command
 - Insert – you can change (edit) the text. You the esc key to go back to command mode
- Popular alternatives to vim – nano (easy), emacs (hard), etc.

diff

- Comparing files for differences (diff)

```
jeec@ix-dev: ~ 64$ cat 20170706-shopping.list
```

```
1 bottle of milk
```

```
2 granny smith apples
```

```
10 hot house tomatoes
```

```
1 six-pack of Coca Cola
```

```
jeec@ix-dev: ~ 65$ cat shopping.list
```

```
1 bottle of milk
```

```
2 granny smith apples
```

```
3 hot house tomatoes
```

```
1 six-pack of Coca Cola
```

```
jeec@ix-dev: ~ 63$ diff shopping.list 20170706-shopping.list
```

```
3c3
```

```
< 3 hot house tomatoes
```

```
---
```

```
> 10 hot house tomatoes
```

diff

```
$ cat file1.txt
this is the original text
line2
line3
line4
happy hacking !

$ cat file2.txt
this is the original text
line2
line4
happy hacking !
GNU is not UNIX

$ diff file1.txt file2.txt
2,3C2
< line2
< line3
---
> line2
5a5
> GNU is not UNIX
```

Compression

- Tar – Tape archive (long long time ago)
- Creating a “tarball” – not necessarily compressed
- e.g., `tar -cvf example.tar /home/example`
 - c – create a new tar archive
 - v – verbose (print information)
 - f – create an archive file
- `tar -xvf example.tar`
 - x – extract the files in the tarball
- gzip can be used to compress the tarball
- e.g., `tar -cvzf example.tar.gz /home/example`
 - z – compress the archive with gzip
- Other options
 - j – use bzz compression (typically higher compression than gzip)
 - t – list the contents of the tarball

Environment variables

- Some variables that provides information on the settings

```
jeec@ix-dev: ~ 68$ env
```

```
SHELL=/bin/bash
```

```
PWD=/home/users/jeec
```

```
MANPATH=/local/share/man:/usr/share/man
```

```
PATH=/home/users/jeec/src/tex/bin/x86_64-linux:/home/users/jeec/bin:/local/bin:/bin:/usr/bin:/sbin:/usr/sbin:/usr/ucb:.
```

```
...
```

```
jeec@ix-dev: ~ 69$ echo $PATH
```

```
/home/users/jeec/src/tex/bin/x86_64-linux:/home/users/jeec/bin:/local/bin:/bin:/usr/bin:/sbin:/usr/sbin:/usr/ucb:
```


man (manual)

- Provides a description of various commands, system calls, and library functions (C standard library)
- man is your friend!

Misc

- which
- locate



Live demonstration of using Linux (if time allows)

Homework 1 & Lab 1

Exercise in using Bitbucket and git

During Lab 1, the TA will help you in setting up Bitbucket and using git

Lab 1 will **NOT** be **graded**

Homework 1 **WILL** be **graded** - be sure to follow the instructions carefully - because there is very little to do, any mistakes will result in fairly large penalty