CS6

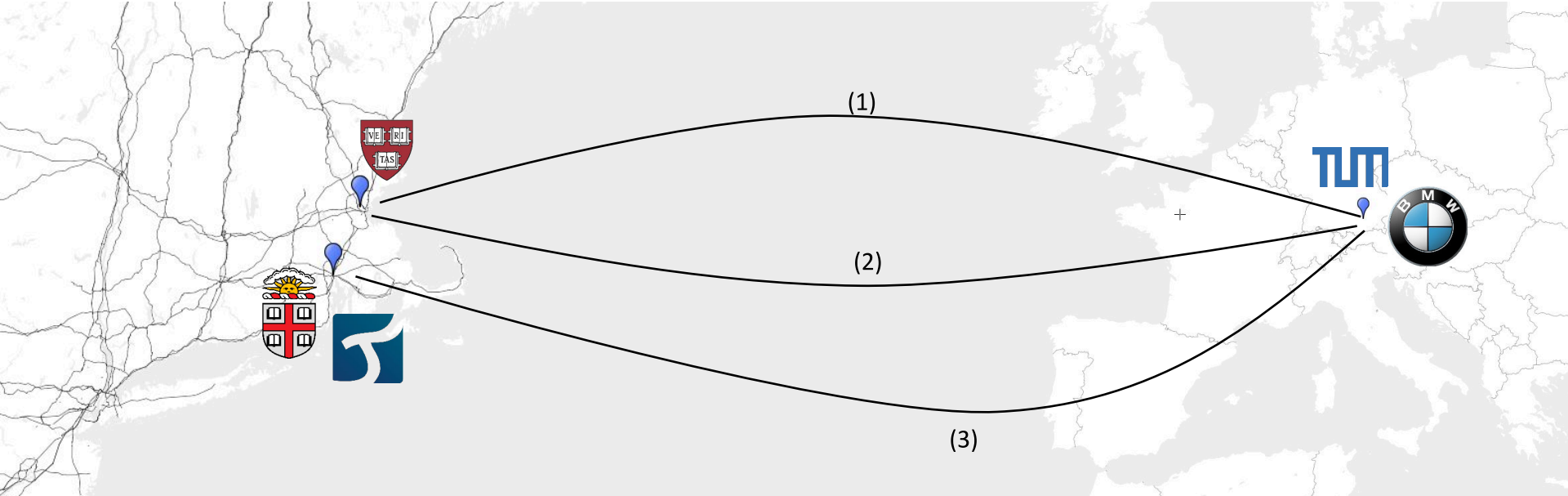
Practical System Skills

Fall 2019 edition

Leonhard Spiegelberg
lspiegel@cs.brown.edu

00.01 About me

- 3rd PhD student in the database management group
- Previous stops: TU Munich, Harvard, BMW
- Current research project: tuplex.cs.brown.edu



00.02 Your CS6 TAs

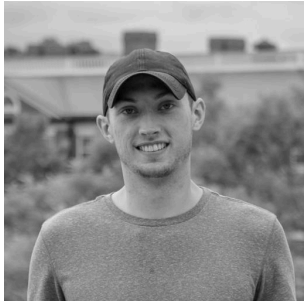
Samuel
HTA

Anina
UTA

Raymond
UTA

Hersh
UTA

Lena
UTA



James
UTA

00.03 Who should take this course?

- Want to learn terminal skills
- Want to learn how to effectively work in an UNIX environment
  - Locally
  - Remotely
- Best practices for collaborative development
- The “secret” sauce to build something fast
- You have already taken an intro to programming class or course

00.04 Today's lecture

- Welcome to Unix ~ 45min
- 5min break
- Logistics ~ rest

Homework 0 out today!

Due: Tue 10th Sep, 4pm

Website: https://cs.brown.edu/courses/cs0060

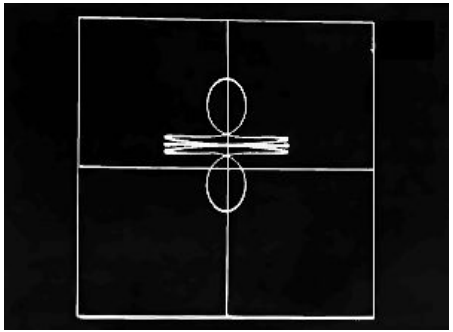
01 Welcome To Unix

CS6 Practical System Skills

Fall 2019

Leonhard Spiegelberg *lspiegel@cs.brown.edu*

01.01 Why Unix?



This game started it all...

Unix history

1969 Ken Thompson wants to run his space game more economically, decides to create UNICS

1970 Name changes to UNIX, because of inspiration from MULTICS

1973 Dennis Ritchie introduces UNIX to a broader audience at an ACM symposium

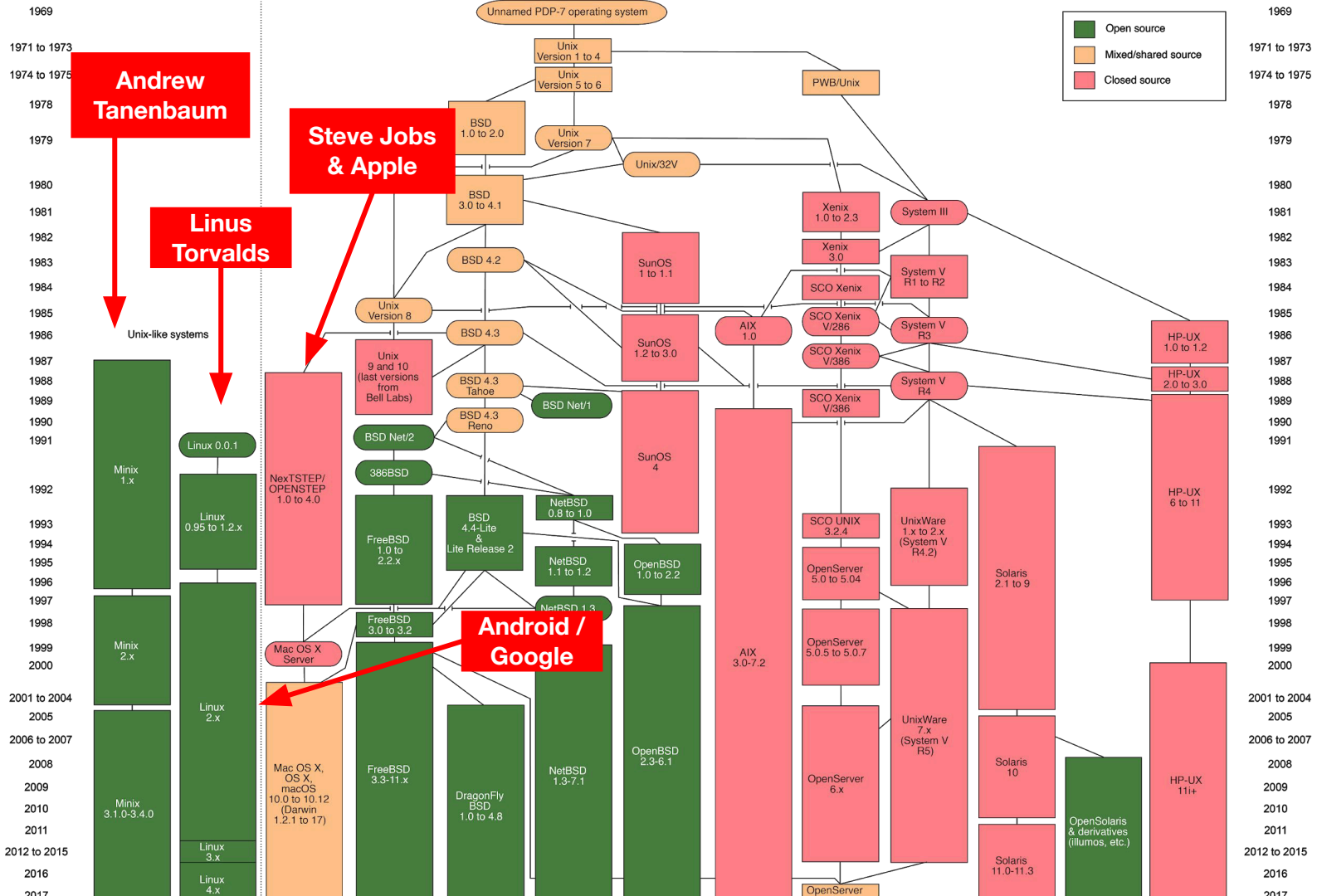
1974 A copy of UNIX from Bell labs arrives at UC Berkeley, the start of the Berkeley Software Distribution

Dennis Ritchie
Turing Award '83

Kenneth Thompson
Turing Award '83

"...the number of UNIX installations has grown to 10, with more expected..."

- Dennis Ritchie and Ken Thompson, June 1972


Andrew Tanenbaum
Linus Torvalds
Steve Jobs & Apple
Android / Google
Unix-like systems
Open source
Mixed/shared source
Closed source
1969
1971 to 1973
1974 to 1975
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001 to 2004
2005
2006 to 2007
2008
2009
2010
2011
2012 to 2015
2016
2017
Unnamed PDP-7 operating system
Unix Version 1 to 4
Unix Version 5 to 6
PWB/Unix
BSD 1.0 to 2.0
Unix Version 7
Unix/32V
BSD 3.0 to 4.1
Xenix 1.0 to 2.3
System III
Xenix 3.0
BSD 4.2
SunOS 1 to 1.1
System V R1 to R2
SCO Xenix
Unix Version 8
BSD 4.3
SunOS 1.2 to 3.0
AIX 1.0
SCO Xenix V/286
System V R3
HP-UX 1.0 to 1.2
SCO Xenix V/386
Unix 9 and 10 (last versions from Bell Labs)
BSD 4.3 Tahoe
BSD Net/1
System V R4
HP-UX 2.0 to 3.0
BSD 4.3 Reno
Minix 1.x
Linux 0.0.1
NexTSTEP/ OPENSTEP 1.0 to 4.0
BSD Net/2
386BSD
SunOS 4
Linux 0.95 to 1.2.x
FreeBSD 1.0 to 2.2.x
BSD 4.4-Lite & Lite Release 2
NetBSD 0.8 to 1.0
HP-UX 6 to 11
SCO UNIX 3.2.4
UnixWare 1.x to 2.x (System V R4.2)
Solaris 2.1 to 9
OpenServer 5.0 to 5.04
NetBSD 1.1 to 1.2
OpenBSD 1.0 to 2.2
Minix 2.x
FreeBSD 3.0 to 3.2
NetBSD 1.3
OpenServer 5.0.5 to 5.0.7
AIX 3.0-7.2
Mac OS X Server
Linux 2.x
UnixWare 7.x (System V R5)
Solaris 10
Minix 3.1.0-3.4.0
Mac OS X, OS X, macOS 10.0 to 10.12 (Darwin 1.2.1 to 17)
FreeBSD 3.3-11.x
DragonFly BSD 1.0 to 4.8
NetBSD 1.3-7.1
OpenBSD 2.3-6.1
OpenServer 6.x
OpenSolaris & derivatives (illumos, etc.)
HP-UX 11i+
Linux 3.x
Solaris 11.0-11.3
Linux 4.x
OpenServer

01.01 Why Unix?

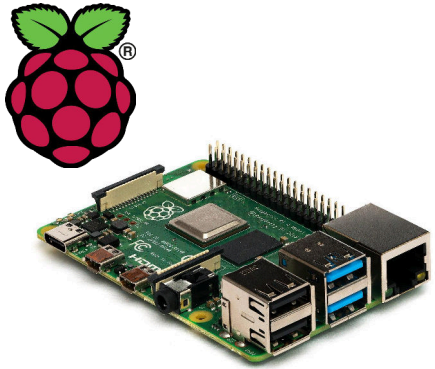
Though the original UNIX system is basically not used any more, its derivatives are prevalent. We often refer to **UNIX**, but in fact mean **derivatives** of it.

⇒ “UNIX” like environment,

⇒ ***NIX** systems

BSD = Berkeley Software Distribution

01.01 Conclusion: Unix is everywhere!



embedded devices

smartphones & other handheld devices

servers

These are the exciting devices for this course!

01.02 GNU/Linux

Linus Torvalds
Linux kernel

Richard Stallman
GNU project

⇒ In this course we'll be learning how to effectively use GNU/Linux

⇒ Most of the tools you'll learn in this class are related to these two persons in some way

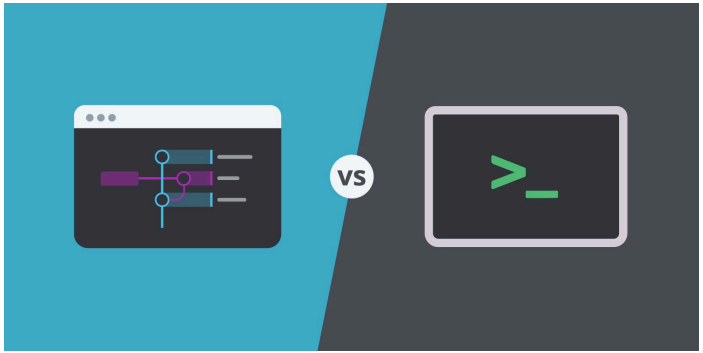
01.02 Interesting devices running Linux

- Nuclear submarines
- The New York Stock Exchange
- Japanese High Speed Trains (Shinkansen)
- Washing Machines
- Smart fridges
- Amazon Kindle
- Self Driving Cars
- Large Hadron Collider
- SteamOS
- Cameras (DSLRs, mirror-less cameras)

⇒ Once you mastered CS6 you can theoretically work with all these devices!

01.03 Interacting with a computer

GUI = Graphical User Interface



CLI = Command Line Interface

⇒ In this course we'll be learning how to effectively use GNU/Linux

⇒ Interaction with a CLI often happens in a REPL (read-evaluate-print-loop)

01.03 CLI and REPL

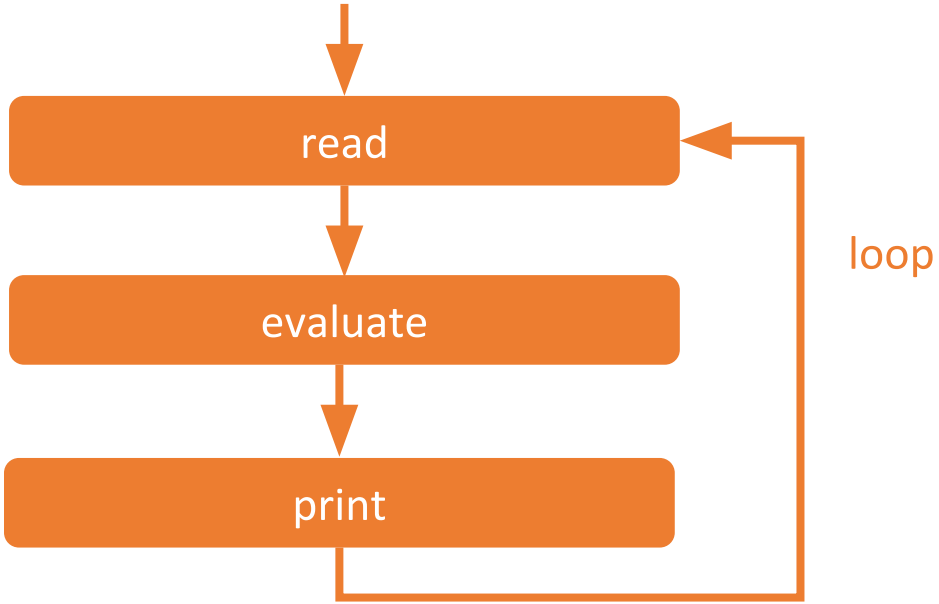
(enter a command)

> **read** a command

> **evaluate** the command

> **print** the results of the command

> **loop** back to start



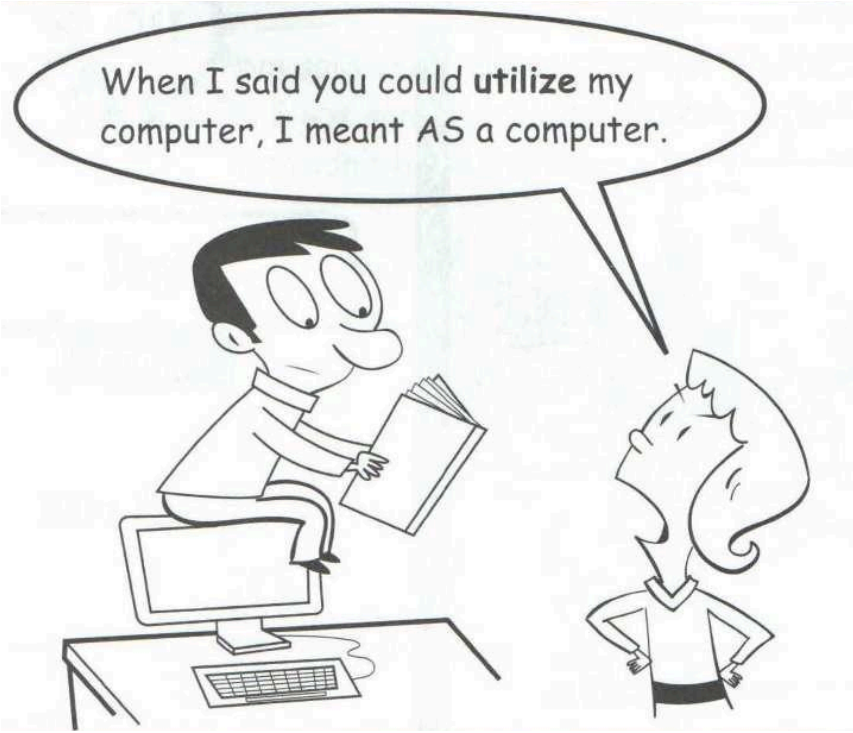
CLI = Command Line Interface

REPL = read-evaluate-print-loop

01.03 Working with a CLI

Learning how to work with a CLI is like learning a new language

⇒ practice, vocabulary and grammar matters



Why is this fun?

01.03 Working with a CLI

Starting a terminal and typing a first command

4. Default (bash)

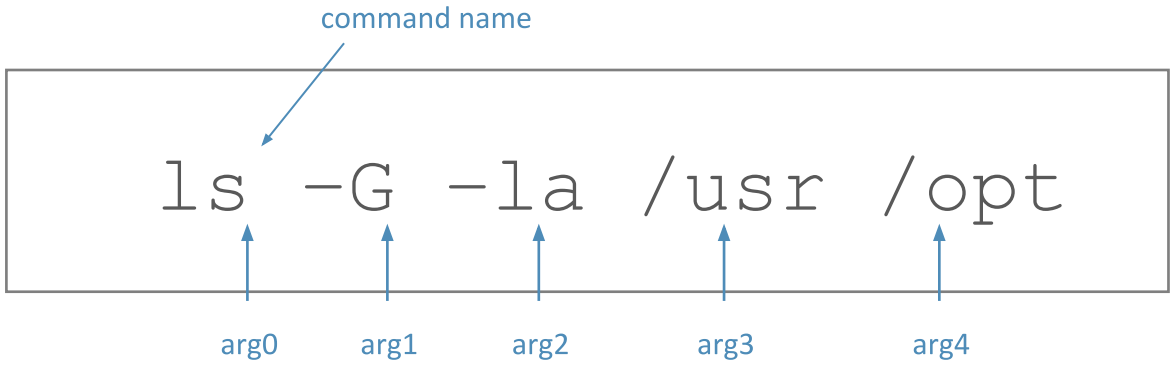
```
Leonhards-iMac:~ leos$ ls -G -la /usr /opt
```

4. Default (bash)

```
Leonhards-iMac:~ leos$ ls -G -la /usr /opt
/opt:
total 0
drwxr-xr-x   3 root  wheel    96 Oct  1  2018 .
drwxr-xr-x  36 root  wheel  1152 Jun 19 16:30 ..
drwxr-xr-x  26 root  wheel   832 Oct  1  2018 intel

/usr:
total 0
drwxr-xr-x@   9 root  wheel    288 Sep 21  2018 .
drwxr-xr-x   36 root  wheel   1152 Jun 19 16:30 ..
drwxr-xr-x  971 root  wheel  31072 Jun 10 13:04 bin
drwxr-xr-x  307 root  wheel   9824 Jun 10 13:04 lib
drwxr-xr-x  250 root  wheel   8000 Jun 10 13:04 libexec
drwxr-xr-x   17 root  wheel    544 Mar 20 11:39 local
drwxr-xr-x  239 root  wheel   7648 Jun 10 13:04 sbin
drwxr-xr-x   46 root  wheel   1472 Mar 20 11:28 share
drwxr-xr-x    5 root  wheel    160 Sep 21  2018 standalone
Leonhards-iMac:~ leos$
```

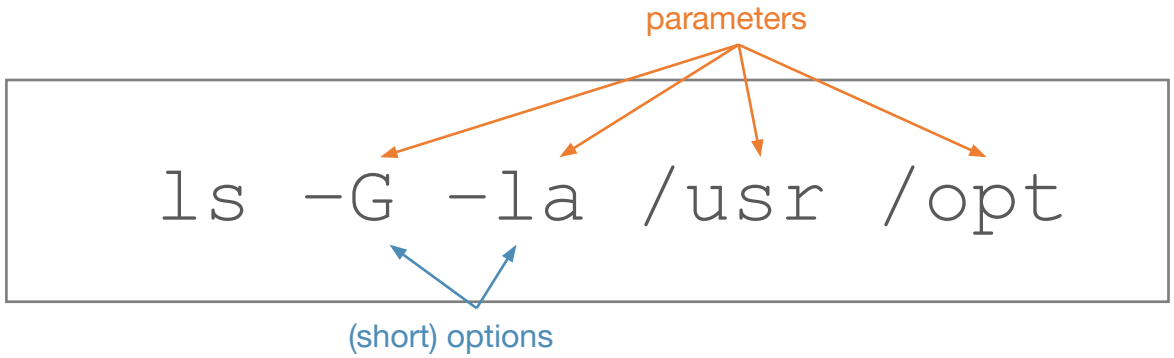
01.04 Commands



A ***command*** is made up of an array of strings called ***arguments*** which are separated by (white)space

- Argument 0 is the command name
- Argument 1 the word after
- ...

01.04 Commands



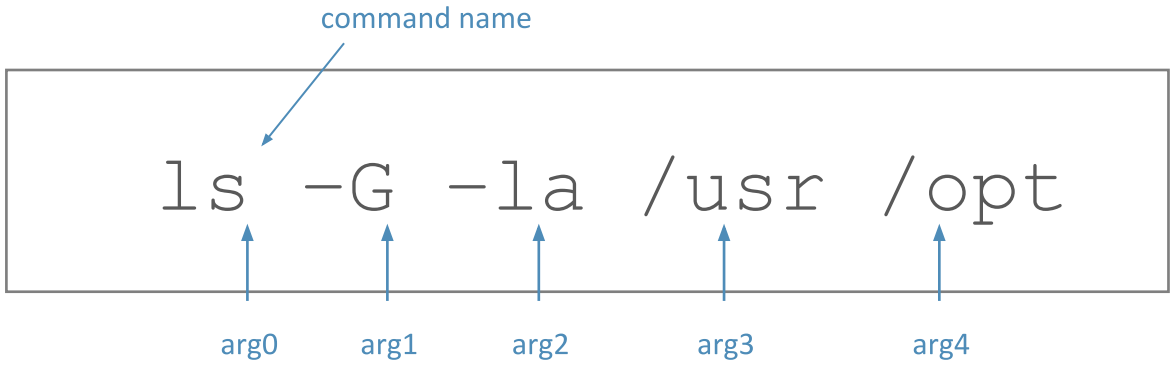
The behavior of a command can be changed using ***options***

- option: argument with a leading `-`
- Two flavors: *Short option:* `-v` *Long option:* `--verbose`

Arguments that provide information to the program are also referred to as ***parameters***

- E.g. an option with a parameter. Example: `-o output.txt`
- Or an argument which serves as parameter to the program. Example: `ls -G`

01.04 Commands



Short options may be concatenated, i.e. `ls -l -a -G` is the same as `ls -laG`

Long options often exhibit a `--key=value` syntax or `--key value` syntax

01.05 How to use command x?

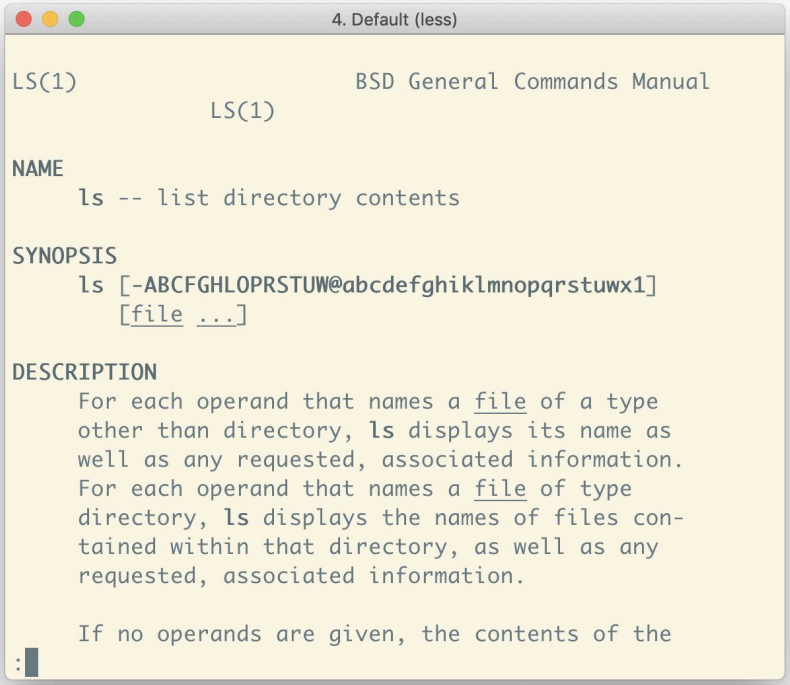
Unix has a special command `man` which brings up a manual on how to use command `x`

E.g., to look at the manual for `ls` use

```
man ls
```

Tip:

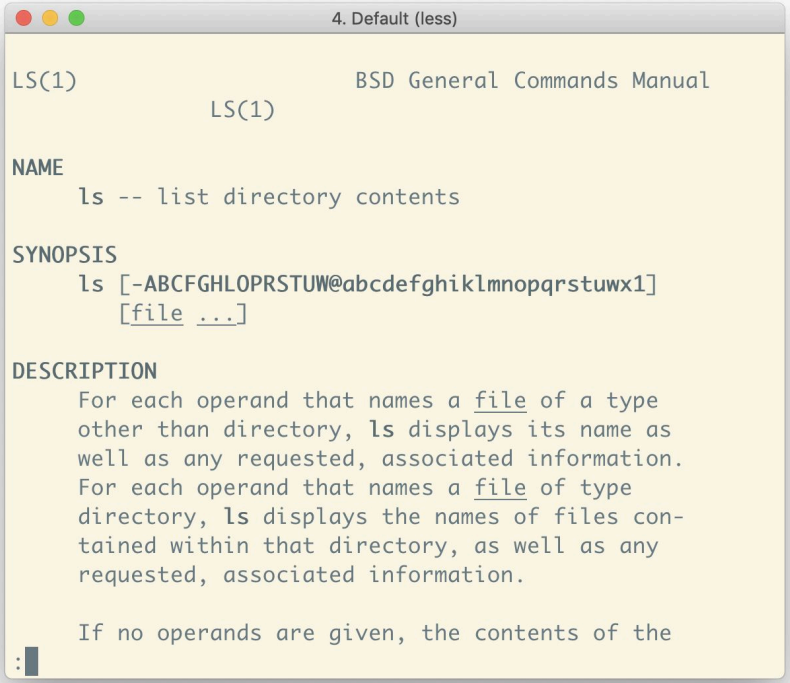
Navigate the manual using your arrow keys ↑↓. To exit the manual program `man`, type `q`.



01.06 How to read man pages?

Synopsis shows how to use the program

- `[argument]` means `argument` is optional
- `a | b` means you either can write `a` or `b`
- `...` means you can repeat the previous argument multiple times (`...` is called ellipsis or ellipsis operator)
- the description section explains in detail what each option does and what each parameter means



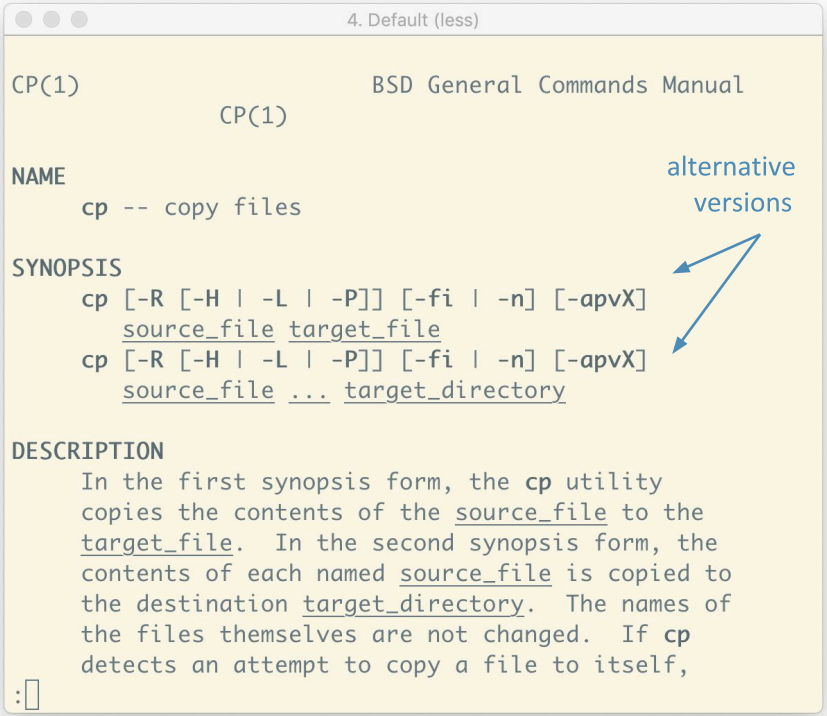
01.06 How to read man pages?

Another example: `cp`

valid syntax	invalid syntax
`cp file.txt copy.txt`	`cp file.txt`
`cp -RH folder/ dest/`	`cp -RHL`
`cp -a folder/ dest/`	
`cp -pPR f.txt a.0`	
`cp -RHL a.txt b.txt`	
`cp -n a.txt b.txt dest/`	

Hint:

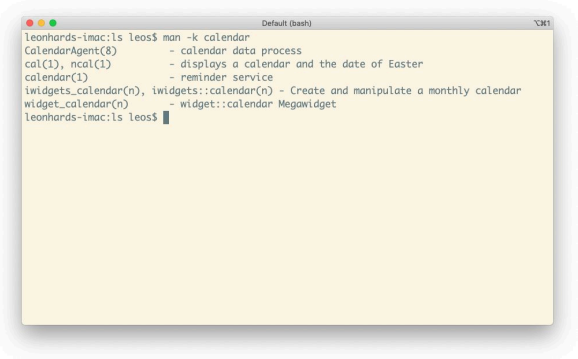
Order of options does not matter, but order of required arguments does! They are therefore also called ***positional arguments.***



01.06 More about man

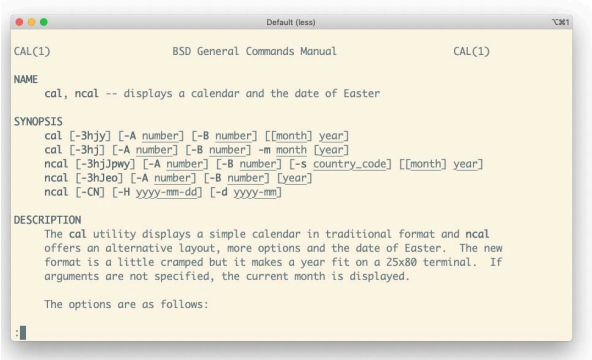
`man` can be also used to search for a command

```
man -k calendar
```



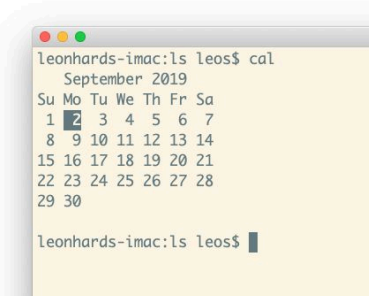
(1) search all man pages via

`man -k calendar`



(2) display help via

`man cal`



(3) run command

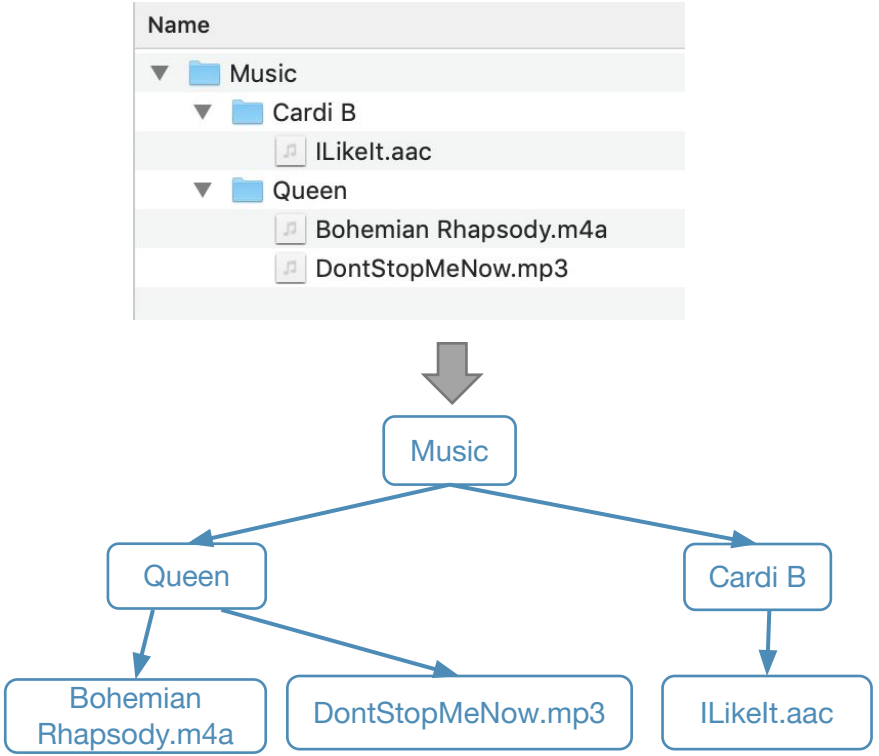
`cal`

01.07 Navigating the file system

All data is organized into files. These files form a tree, the file system.

5 types of files in Unix:

- ***ordinary files*** ⇐ **today**
- ***directories*** ⇐ **today**
- *(symbolic) links*
  [also called *symlinks*]
- *device files*
  [also called *special files*]
- *FIFO files*

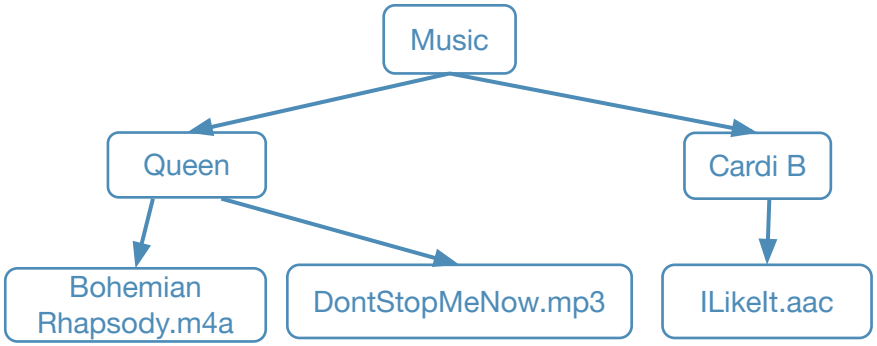


01.07 Absolute file paths

Each file has a ***unique absolute file path*** identifying its location in the file system.

An ***absolute (file) path*** is created by following the directory structure from the root of the file system `/` to the file.

An absolute (file) path <u>always</u> starts with `/` and must consist of 0 to n directories and at most one file at the end.



<u>Examples:</u>

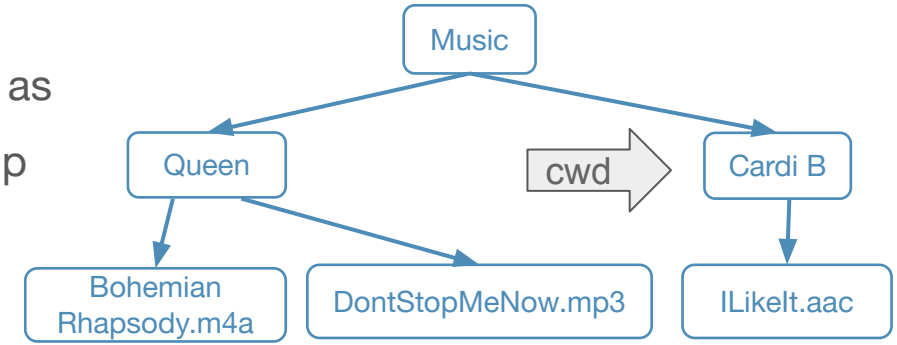
file separator

```
/
/Music/Queen/DontStopMeNow.mp3
/Music/Queen
```

are all absolute paths

01.07 Relative file paths

Relative file paths are paths relative to the ***current working directory*** (often abbreviated as ***cwd***) which like absolute file paths are made up of directories and at most one file at the end joined together by



Three special symbols to create relative paths:

.	current directory	
. .	parent directory	
~	home directory	the directory where the shell is initially at

Examples:

.	../Queen
./	../Queen/
..	./../Queen/DontStopMeNow.mp3

01.07 More on file paths

Though undefined, a path with a ***trailing slash*** `/` often means that the path is a directory. It may be used by a command to refer to the contents of a directory then.

Some characters need to be ***escaped*** on paths, this is done using a backslash `\`

`Cardi B` ⇒ `Cardi\ B`

General advice: Avoid using special characters when composing file names.
⇒ Stick to lowercase/uppercase `a-z` (ABC) “`.`”, “`_`” and “`-`”

```
this_is-a-good-name-123.txt
```

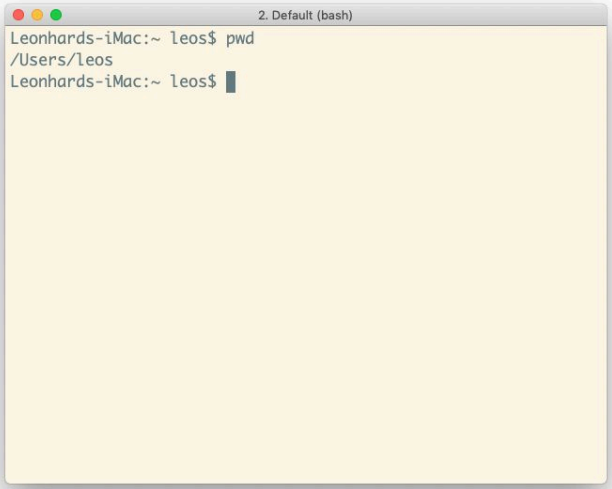
```
don't name your files like this!.txt
```

01.07 More on file paths

Given a path, a common operation is to retrieve the name of the file the path is referring to (i.e. the suffix after the last file separator `/`). This is called the ***basename*** of the path. E.g.

- the basename of `/Home/Queen/Dontstopmenow.mp3`is `Dontstopmenow.mp3`
- the basename of `.` is `.`, the basename of `..` is `..`, the basename of ~ is ~
- the basename of `../../hello.mp3` is `hello.mp3`
- the basename of `Music/` is the empty string!
- the basename of `Music/Queen` is `Queen`

01.07 Navigating the file system - back to the Terminal!



`pwd`	**p**rint **w**orking **d**irectory	print (absolute) path of the current directory we are in
`ls`	**l**i**s**t	list files in the current directory
`cd`	**c**hange **d**irectory	change shell's working directory, i.e. go to some directory

01.08 change directory

`cd`	go to home directory
`cd ~`	go to home directory
`cd ~user`	go to home directory of user
`cd .`	go to current directory (no change)
`cd ..`	go to parent directory
`cd -`	switch to previous directory
`cd path`	go to path (can be a relative or absolute path)

Tip:

This is one of the MOST used commands in a terminal.
It is like the “to be” verb in any language, master it!

01.09 list

`ls`	list all visible files in cwd
`ls -a`	list all files in cwd
`ls [file ...]`	run list for all given files, list them separately

```
.
├── .hidden_file
├── .hidden_folder
│   ├── hiding_penguins.txt
│   └── penguins.txt
├── hello.txt
└── world.csv
```

<u>Hidden files:</u>

UNIX defines files whose basename starts with . as hidden files. I.e. files/folders which start with . will not be displayed, unless the option `-a` in `ls` is used.

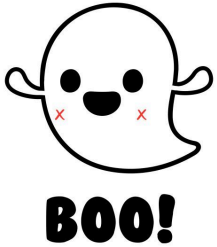
01.09 list - examples

```
ls
ls -a
ls hello.txt
ls .hidden_folder
ls wonders.txt
ls .hidden_folder .
ls penguins.txt hello.txt
ls .hidden_file
ls ./.
```

```
.
├── .hidden_file
├── .hidden_folder
│   ├── hiding_penguins.txt
│   └── penguins.txt
├── hello.txt
└── world.csv
```

If the basename starts with . it's a hidden file!

Uncover it using ls -a



01.09 list - examples

```
.
├── .hidden_file
├── .hidden_folder
│   ├── hiding_penguins.txt
│   └── penguins.txt
├── hello.txt
└── world.csv
```

	result
`ls`	`hello.txt  world.csv`
`ls -a`	`.  ..  .hidden_file  .hidden_folder  hello.txt  world.csv`
`ls hello.txt`	`hello.txt`
`ls .hidden_folder`	`hiding_penguins.txt  penguins.txt`
`ls wonders.txt`	`ls: wonders.txt: No such file or directory`

`ls .hidden_folder .`	`.:`
	`hello.txt  world.csv`
	`.hidden_folder:`
	`hiding_penguins.txt  penguins.txt`

`ls penguins.txt hello.txt`	`ls: penguins.txt: No such file or directory` `hello.txt`
`ls .hidden_file`	`.hidden_file`
`ls ./.`	`hello.txt  world.csv`

01.09 list - more options

`ls -G`	display folders/files with different colors
`ls -l`	list files in listmode (i.e. with extended attributes)
`ls -F`	append / to folders
`ls -d`	avoid listing paths in separate directory aggregates

01.09 list - more options

```
.
├── .hidden_file
├── .hidden_folder
│   ├── hiding_penguins.txt
│   └── penguins.txt
├── hello.txt
└── world.csv
```

command	result
`ls -aG`	`.  ..  .hidden_file  .hidden_folder  hello.txt  world.csv`
`ls -aF`	`./  ../  .hidden_file  .hidden_folder/  hello.txt.` `world.csv`
`ls -ad . .hidden_folder`	`.  .hidden_folder`

`ls -l`	`total 0`
	`-rw-r--r--@ 1 leos  staff  0 Jul 30 16:46 hello.txt`
	`-rw-r--r--@ 1 leos  staff  0 Jul 30 16:46 world.csv`

01.09 working with the list command

Often you can achieve the same thing with different strategies, find out which strategy works the best for you!

Example: How to get an overview of the contents of a directory?

no d in the attributes, directory contains two files.

```
leonhards-imac:ls leos$ ls -l
total 0
-rw-r--r--@ 1 leos  staff  0 Jul 30 16:46 hello.txt
-rw-r--r--@ 1 leos  staff  0 Jul 30 16:46 world.csv
leonhards-imac:ls leos$
```

no names with trailing /, directory contains two files.

```
leonhards-imac:ls leos$ ls -F
hello.txt       world.csv
leonhards-imac:ls leos$
```

everything has the same color, directory contains two files.

```
leonhards-imac:ls leos$ ls -G
hello.txt world.csv
leonhards-imac:ls leos$
```

5 min break...

… then logistics!

02 Logistics

CS6 Practical System Skills

Fall 2019

Leonhard Spiegelberg *lspiegel@cs.brown.edu*

02.00 Course Overview

09/04 - 10/03 Terminal skills
Files, Permissions, Streams, Pipes, Scripts, SSH, Processes, Text Processing, Regex

10/08 – 10/17 Websites & Git
HTML, CSS, version control

10/22 Midterm I + start of final projects

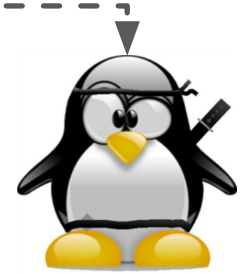
10/24 - 11/12 Web backends
Flask, Python, MongoDB, MySQL

11/14 – 11/21 Data analytics
DataFrames, data visualization

11/22 Midterm II

12/3 – 12/5 Selected Topics

12/15 Final project due



02.01 Homeworks

Weekly homeworks, HW0 out today!

Due: next Tuesday 9/10, 4pm

HW0 bears no credit,
but you are required to submit it in order to register for the course!

no late days

If you submit **0-24h** late, we will deduct **25%**. No grading for a homework that is submitted more than 24h late.

I.e. all homeworks received after a Wed, 4pm will receive a score of 0.

02.02 Exams

Two midterms (cumulative)

Midterm I: **10/22**

Midterm II: **11/22**

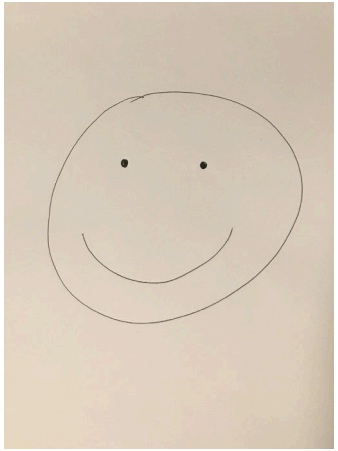
In each midterm you will be allowed one US-letter sized sheet with <u>your</u> handwritten notes.

No phones, notebooks, or magnifying devices allowed!

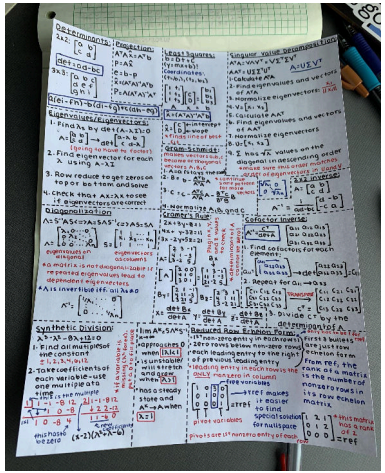
02.03 Cheatsheets

Each cheatsheet must contain your **own** personal, handwritten notes.

It is ok to collaborate on the content of a cheatsheet,
i.e. preparing together for a midterm.



optimistic
approach



defensive approach

02.04 Final Projects

Groups of 3, advised by a TA.

Goal: Build & deploy something!

Start: After Midterm I

02.05 Grading

Mandatory S/NC ⇒ pass or fail

in order to pass ~70%

Breakdown:

Homeworks	**50 %**
Midterm 1	**15 %**
Midterm 2	**15 %**
Final project	**20 %**

02.06 Hours

Lecture: Tue / Thu 4pm-5:20pm in CIT477

Labs: Tue 8pm-10pm in CIT 201 (first lab next week on Tue, 9/10)

Office hours:

Leonhard: Tue / Thu 3pm-4pm in CIT249

TAs: Check the course website & calendar. Will be posted tonight.

Tip: Subscribe to the CS6 Calendar @
https://cs.brown.edu/courses/cs0060/hours.html

02.07 Resources

Piazza https://piazza.com/class/jzdb5qijlc1594

Optional textbooks & resources https://cs.brown.edu/courses/cs0060/resources.html

All logistics can be found under https://cs.brown.edu/courses/cs0060/index.html

In particular

- Syllabus: https://cs.brown.edu/courses/cs0060/assets/docs/course_syllabus.pdf
- Course Missive: https://cs.brown.edu/courses/cs0060/assets/docs/course_missive.pdf
- Collaboration Policy: https://cs.brown.edu/courses/cs0060/assets/docs/collaboration_policy.pdf

02.08 HW0 preview

Expected time: ~ 0.5-3h

Helps you to get familiar with course procedures

- Gradescope for homework handin
- Piazza for exciting discussions with the CS6 community

Setups SSH to log in to department machines

Gets course environment ready

Note: In order to register for CS6 you have to submit HW0!

Available now on the course website!

02.09 Questions?

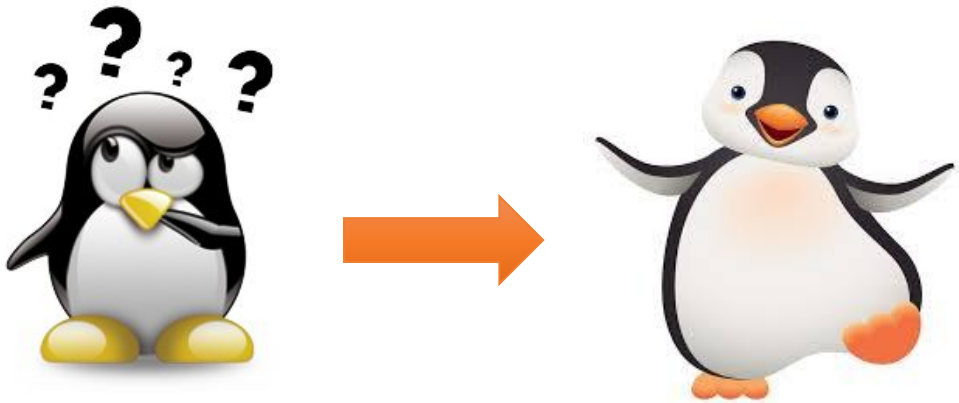
Any course related questions may be mailed to

lspiegel@cs.brown.edu

cs0060tas@cs.brown.edu

cs0060headtas@cs.brown.edu

or asked on **Piazza**.



We are here to help you!

End of lecture.

Next class: Tue, 4pm-5:20pm @ CIT 477