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

Homework 0 out today!

Due: Tue 10th Sep, 4pm
Website: https://cs.brown.edu/courses/cs0060
Welcome To Unix
...the number of UNIX installations has grown to 10, with more expected...

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

- Dennis Ritchie and Ken Thompson, June 1972
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!
⇒ 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!
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).
(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?
Starting a terminal and typing a first command
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
- ...
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
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
Another example: `cp`

<table>
<thead>
<tr>
<th>valid syntax</th>
<th>invalid syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>cp file.txt copy.txt</code></td>
<td><code>cp file.txt</code></td>
</tr>
<tr>
<td><code>cp -RH folder/ dest/</code></td>
<td><code>cp -RHL</code></td>
</tr>
<tr>
<td><code>cp -a folder/ dest/</code></td>
<td></td>
</tr>
<tr>
<td><code>cp -pPR f.txt a.0</code></td>
<td></td>
</tr>
<tr>
<td><code>cp -RHL a.txt b.txt</code></td>
<td></td>
</tr>
<tr>
<td><code>cp -n a.txt b.txt dest/</code></td>
<td></td>
</tr>
</tbody>
</table>

**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
   ```bash
   man -k calendar
   ```
2. display help via
   ```bash
   man cal
   ```
3. run command
   ```bash
   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:

1. **ordinary files** ⇐ today
2. **directories** ⇐ today
3. **(symbolic) links** [also called symlinks]
4. **device files** [also called special files]
5. **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 **always** starts with `/` and must consist of 0 to n directories and at most one file at the end.

**Examples:**

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

are all absolute 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:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.</td>
<td>current directory</td>
</tr>
<tr>
<td>../</td>
<td>parent directory</td>
</tr>
<tr>
<td>~</td>
<td>home directory the directory where the shell is initially at</td>
</tr>
</tbody>
</table>

Examples:

- . ..:/Queen
- ./ ..:/Queen/
- .. ./../Queen/DontStopMeNow.mp3
Though undefined, a path with a \textit{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 \textit{escaped} on paths, this is done using a backslash \\

Cardi B $\Rightarrow$ Cardi\ B

General advice: Avoid using special characters when composing file names. $\Rightarrow$ Stick to lowercase/uppercase \(a-z\) (ABC) ".", ",_" and ",-"

\begin{itemize}
\item \texttt{this_is-a-good-name-123.txt}
\item \texttt{don't name your files like this!.txt}
\end{itemize}
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`
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>pwd</strong></td>
<td>print working directory</td>
</tr>
<tr>
<td><strong>ls</strong></td>
<td>list files in the current directory</td>
</tr>
<tr>
<td><strong>cd</strong></td>
<td>change directory</td>
</tr>
</tbody>
</table>

**print (absolute) path of the current directory we are in**

**list files in the current directory**

**change shell’s working directory, i.e. go to some directory**
01.08 change directory

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>cd</code></td>
<td>go to home directory</td>
</tr>
<tr>
<td><code>cd ~</code></td>
<td>go to home directory</td>
</tr>
<tr>
<td><code>cd ~user</code></td>
<td>go to home directory of user</td>
</tr>
<tr>
<td><code>cd .</code></td>
<td>go to current directory (no change)</td>
</tr>
<tr>
<td><code>cd ..</code></td>
<td>go to parent directory</td>
</tr>
<tr>
<td><code>cd -</code></td>
<td>switch to previous directory</td>
</tr>
<tr>
<td><code>cd path</code></td>
<td>go to path (can be a relative or absolute path)</td>
</tr>
</tbody>
</table>

**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 files:

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 ./.
```

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

Uncover it using ls -a
## 01.09 list - examples

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ls</code></td>
<td>hello.txt  world.csv</td>
</tr>
<tr>
<td><code>ls -a</code></td>
<td>.  .. .hidden_file .hidden_folder  hello.txt  world.csv</td>
</tr>
<tr>
<td><code>ls hello.txt</code></td>
<td>hello.txt</td>
</tr>
<tr>
<td><code>ls .hidden_folder</code></td>
<td>hiding_penguins.txt  penguins.txt</td>
</tr>
<tr>
<td><code>ls wonders.txt</code></td>
<td>ls: wonders.txt: No such file or directory</td>
</tr>
<tr>
<td><code>ls .hidden_folder .</code></td>
<td>.:                     hello.txt  world.csv</td>
</tr>
<tr>
<td></td>
<td>.hidden_folder:         hiding_penguins.txt  penguins.txt</td>
</tr>
<tr>
<td><code>ls penguins.txt</code></td>
<td>ls: penguins.txt: No such file or directory</td>
</tr>
<tr>
<td><code>hello.txt</code></td>
<td>hello.txt</td>
</tr>
<tr>
<td><code>ls .hidden_file</code></td>
<td>.hidden_file</td>
</tr>
<tr>
<td><code>ls ./</code></td>
<td>hello.txt  world.csv</td>
</tr>
</tbody>
</table>
## 01.09 list - more options

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ls -G</code></td>
<td>display folders/files with different colors</td>
</tr>
<tr>
<td><code>ls -l</code></td>
<td>list files in listmode (i.e. with extended attributes)</td>
</tr>
<tr>
<td><code>ls -F</code></td>
<td>append / to folders</td>
</tr>
<tr>
<td><code>ls -d</code></td>
<td>avoid listing paths in separate directory aggregates</td>
</tr>
</tbody>
</table>
## 01.09 list - more options

<table>
<thead>
<tr>
<th>command</th>
<th>result</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ls -aG</code></td>
<td>.     .. hidden_file hidden_folder hello.txt world.csv</td>
</tr>
<tr>
<td><code>ls -aF</code></td>
<td>./    ../ hidden_file hidden_folder/ hello.txt</td>
</tr>
<tr>
<td><code>ls -ad . .hidden_folder</code></td>
<td>. .hidden_folder</td>
</tr>
<tr>
<td><code>ls -l</code></td>
<td>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</td>
</tr>
</tbody>
</table>
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?

- `ls -l` in the attributes, directory contains two files.
- `ls -F` no names with trailing `/`, directory contains two files.
- `ls -G` everything has the same color, directory contains two files.
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

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

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

**10/22** Midterm I + start of final projects

**11/22** Midterm II

**12/3 – 12/5** Selected Topics

**12/15** Final project due
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 your handwritten notes.

No phones, notebooks, or magnifying devices allowed!
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.
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:

<table>
<thead>
<tr>
<th>Assignment</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homeworks</td>
<td>50 %</td>
</tr>
<tr>
<td>Midterm 1</td>
<td>15 %</td>
</tr>
<tr>
<td>Midterm 2</td>
<td>15 %</td>
</tr>
<tr>
<td>Final project</td>
<td>20 %</td>
</tr>
</tbody>
</table>
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!
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