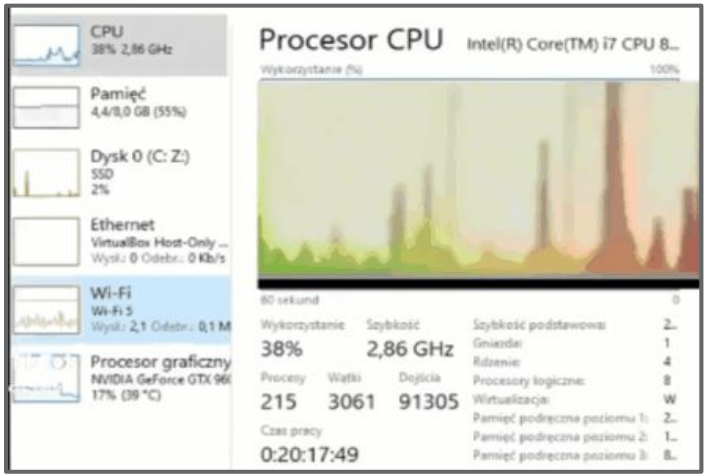
What's happening CEG2350?

Aiden Cox - CEG 2350 Lab Lead
Austin Kellough - CEG 2350 Lab Assistant
Quote of the week:
"Yesterday is history, tomorrow is a mystery, but today is a gift. That is why we call it the present"

How was Lab08?



Sudo-ing the making of sandwiches?
Partitioning?

Can't SSH into your AWS Instance?

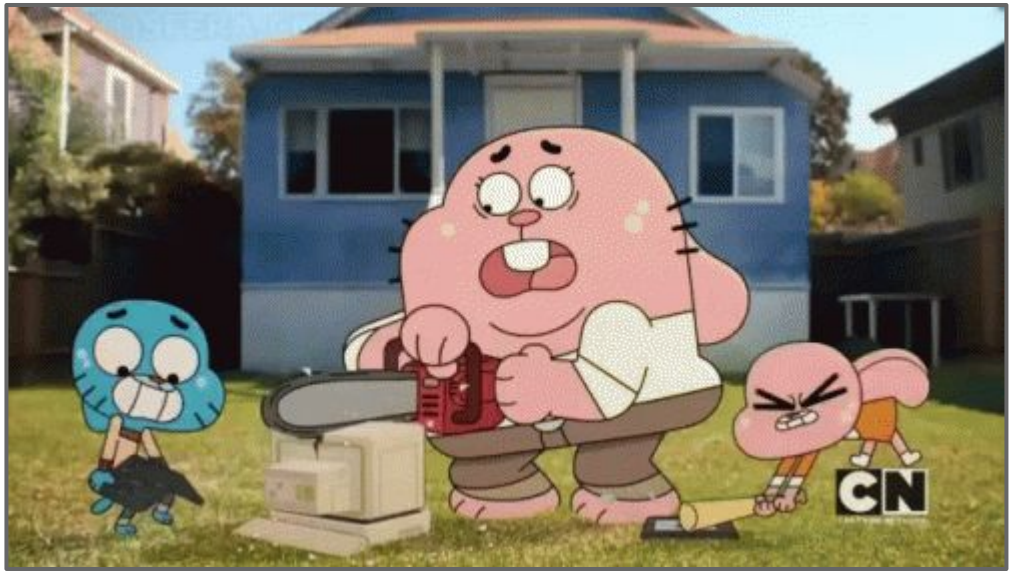
If you cannot `ssh` to your instance anymore, you may have run into one of these scenarios:

- You overwrote the partition table (or partitions) in `xvda` - this would erase your `root` / `/` filesystem.
- You wrote a bad entry in `/etc/fstab` - if the system cannot mount the disk, the boot process will hang and not complete

If you think one of those scenarios happened to you, you'll need to go back to AWS Academy Setup and create a new stack. Once you are in the new instance, don't forget the steps to cloning your GitHub repo:

1. create a new keypair for authentication to GitHub
2. add the public key to your GitHub user settings
3. clone with `ssh`

Midterm Reminder - Nov 4th (Next Tuesday)



Would we like a similar format from last exam? Review heavy this Monday, more lab focused during lab time on Wednesday after the exam?

Beginning Lab09

Lab Instructions: https://pattonsgirl.github.io/CEG2350/Labs/Lab09/Instructions.html

Lab Template:

Feeling comfortable with the course?
Questions from lecture?

Game the System - Part 1

Part 1 - Game the system

Useful commands: `apt` , `which` , `whereis`

Choose a command line game from this site: It's Foss - Top 10 Command Line Games for Linux

- `bastet` is what this lab was tested on

In the answer template, document the game name, how you installed it, where the executable is located, and how to run the game.

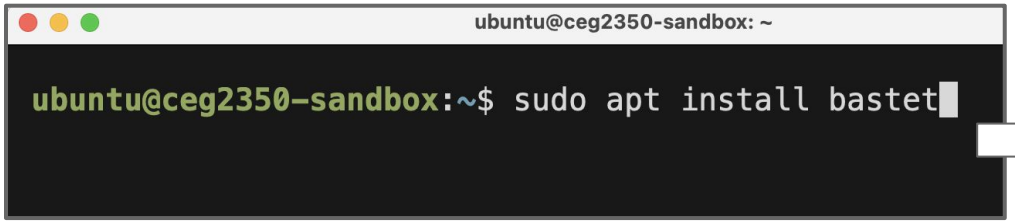
For the remainder of this lab, you will be running this game to practice controlling processes.

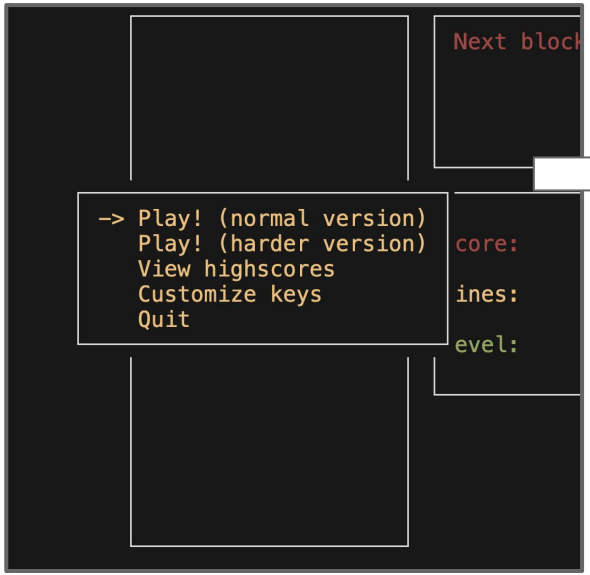
Questions asked for Part 1

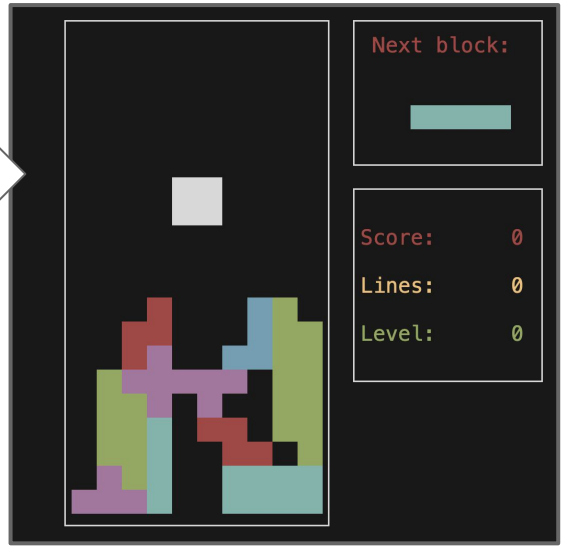
Part 1 - Game the system

- Game name:
- How to install:
- Location of game executable:
- How to run game:

Game the System - Part 1







Any game will do! But I like to play tetris instead of listening to lectures.

Process Control - Part 2

Part 2 - Process control

Create a second `ssh` session to your AWS instance. You should now have two `ssh` sessions to your AWS instance. This will be referred to as Shell A and Shell B in the exercises below - you decide which is A and which is B.

- **Useful Commands:** `pstree`, `ps`, `kill`, `bg`, `job`, `fg`

1. Identify the following `ps` fields:
   - USER / UID, PID, PPID, TTY, STAT, and COMMAND / CMD
2. Craft a `ps` command for processes owned by your user that will show the fields listed above
3. In Shell A, run the game. Using Shell B, run your `ps` command. Provide answers to the following, using Shell B to observe the process statuses:
4. For the game:
   - What is it's process id?
   - What is it's parent process id?
   - What process is the parent process?
5. Use `kill` to kill only the game.
   - Describe what the effect was.
6. [Run the game again] Use `kill` to kill the game and it's parent process.
   - Describe what the effect was.
7. Create a new `ssh` session - Shell C - and run the game again. Watch the processes from the other terminal. Describe what happens, using process knowledge in your description, **if you close / `exit` your connection with Shell C and determine if you can reenter the game (resume the process)**.

Questions asked for Part 2

Process Control - Part 2

1. `ps` field descriptions:
   - USER / UID:
   - PID:
   - PPID:
   - TTY:
   - STAT:
   - COMMAND / CMD:
2. `ps` command:
3. Output of `ps` with two shells, one running the game:

```
Output of command goes here
```

4. For the game:
   - Process id:
   - Parent process id:
   - What is the parent process:
5. `kill` to kill only the game:
   - Describe what the effect was:
6. `kill` to kill the game and it's parent process:
   - Describe what the effect was:
7. Describe what happens if you close / `exit` your connection with Shell C and determine if you can reenter the game (resume the process).
   - Answer:

Answer template for Part 2

Process Control - Part 2

Useful commands and flags for `ps`!

```
PS(1)                    User Commands                    PS(1)

NAME
       ps - report a snapshot of the current processes.

SYNOPSIS
       ps [options]

DESCRIPTION
       ps displays information about a selection of the active
       processes.  If you want a repetitive update of the
       selection and the displayed information, use top
       instead.
```

```
ubuntu@ceg2350-sandbox:~$ ps -u ubuntu -o FD1here,FD2here,3here,...
```

```
-o format
       User-defined  format.   format is a single argument in the
       form of a blank-separated or comma-separated  list,  which
       offers  a  way  to specify individual output columns.  The
       recognized keywords are described in the  STANDARD  FORMAT
       SPECIFIERS  section  below.  Headers may be renamed (ps -o
       pid,ruser=RealUser -o comm=Command) as  desired.   If  all
       column  headers  are  empty (ps -o pid= -o comm=) then the
       header  line  will  not  be  output.   Column  width  will
       increase  as  needed for wide headers; this may be used to
```

Process Control - Part 2

Useful commands and flags for `kill`!

```
    PPID     PID TT       USER       UID CMD
       1     941 ?        ubuntu    1000 /usr/lib/systemd/systemd --user
     941     942 ?        ubuntu    1000 (sd-pam)
     936    1050 ?        ubuntu    1000 sshd: ubuntu@pts/0
    1050    1051 pts/0    ubuntu    1000 -bash
    1415    1471 ?        ubuntu    1000 sshd: ubuntu@pts/1
    1471    1472 pts/1    ubuntu    1000 -bash
    1472    1596 pts/1    ubuntu    1000 bastet
```

Back and Fore - Part 3

Part 3 - back and fore

Questions asked for Part 3

1. Run the game in the foreground.
2. Send a `STOP` signal to suspend it.
3. Use `ps` to confirm the process is still alive, but has been stopped. Provide the line of output that relates to the process.
4. Resume the process in the foreground.
5. Send a `TERMINATE` signal to kill it.
6. Start the game as a background process. Repeat 3 times.
7. Display the output of `jobs` and your custom output format `ps` command
    - If `jobs` is empty, make sure you are running it in the same shell, the controlling terminal, that you created the background jobs.
8. Kill one job.
9. Move one job to the foreground.
10. Describe what happens, using process knowledge in your description, if you close / `exit` your connection with this shell and determine if you can reenter the game (resume the process).

Back and Fore - Part 3

Part 3 - back and fore

Answer template for Part 3

1. Run in the foreground:
2. `STOP` signal to suspend it:
3. Proof of life from `ps` output:

```
insert line here
```

4. Resume in the foreground:
5. `TERMINATE` signal to kill it:
6. Start as a background process:
7. Output of `jobs` from controlling terminal:

```
Insert output here
```

Output of `ps` :

```
Insert output here
```

8. Kill job:
9. Move job to the foreground:
10. Describe what happens, using process knowledge in your description, if you close / `exit` your connection with this shell and determine if you can reenter the game (resume the process).

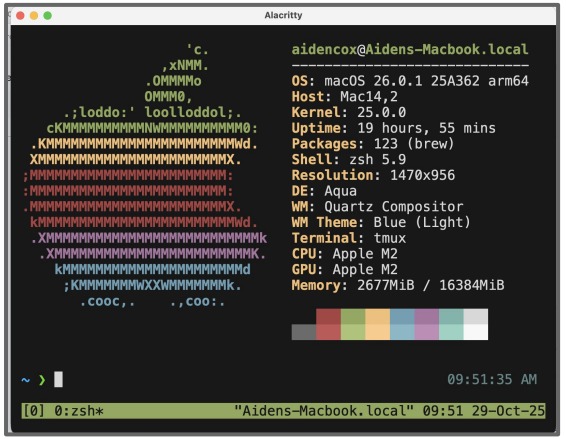
- Answer:

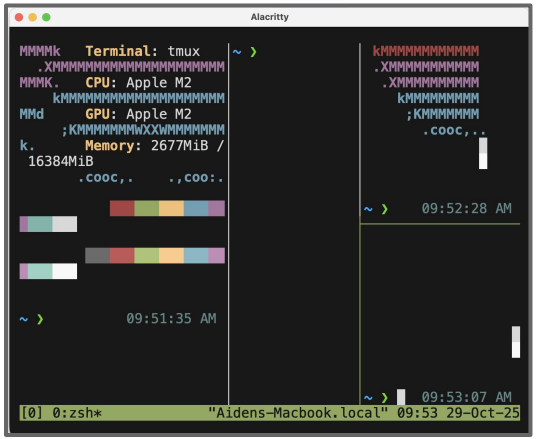
Detach - Part 4

Part 4 - Detach

The perhaps obtuse goal of the previous two exercises is to understand process control, but also to realize that with these methods, your shell connection must remain active. If you end your shell session, the processes attached to it also end. There are tools that allow you to run processes detach from your shell session. This lab will have you use `tmux`, but other tools exist.

1. Fill out the following table of `tmux` commands and keybinds:





1. Create a `tmux` session. Run the game in the session.
2. Leaving your game running, detach from the session.
3. Run your custom formatted `ps` command, showing processes without a controlling terminal
4. Use `tmux` to list sessions
5. Close your `ssh` shell session to your instance, then `ssh` in again. How can you determine if your `tmux` session with your game running is available?
6. Reattach to your `tmux` session running the game.
7. Kill the `tmux` session

Detach - Part 4

Task	Command
Start a new session and set a name	`tmux new -s <session_name>`
List tmux sessions	`tmux ls`
Attach to session by name	`tmux attach -t <session_name>`
Kill session by name	`tmux kill-session -t <session_name>`
Detach from session	Press `Ctrl + b`, then `d`
Split pane horizontally	`Ctrl + b`, then `%`
Split pane vertically	`Ctrl + b`, then `"`
Switch panes	`Ctrl + b`, then arrow keys
Close pane	Exit the shell in that pane (`exit`), or `Ctrl + b`, then `x`

Have a Good Weekend!

Don’t hesitate to reach out and ask questions!
Quote of the week:
“Yesterday is history, tomorrow is a mystery, but today is a gift. That is why we call it the present”