351 Section 1

Welcome to 351!

- First: this isn't 14X any more
  - You can get a lot of help in ways you may not be used to
  - You can work with other people
    - Plagiarism policies are outlined on the website
    - Generally, we place a lot of trust in you, and will revisit that if there's an issue
- This course can feel slow at times in terms of the output you produce
  - Thinking a lot more, writing a lot less
  - Don't worry -- this is normal!
- You will be introduced to a lot of new stuff, so make sure you're taking the time to grasp the fundamentals -- they will serve you for the rest of your time in CS / EE

My role

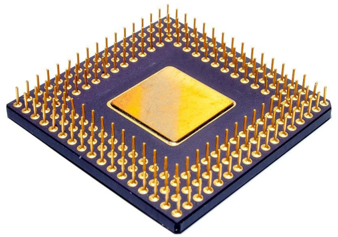
- TAs are the first point of contact!
- We will be monitoring:
    - The message board
    - The email list
    - Our personal emails, if you need to email one of us for some reason
- Office hours are a fantastic resource
    - We literally just hang out and wait for people to ask us questions about pretty much anything 351-related
    - Please come to office hours and ask questions!

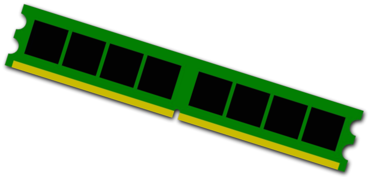
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY

START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY

START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY
START EARLY

```
for (i=0; i<n; i++)
int main(int argc, char* argv[])
fopen("file.dat", "r");
int
char
printf("Hello world");
ret
lea
movl
```



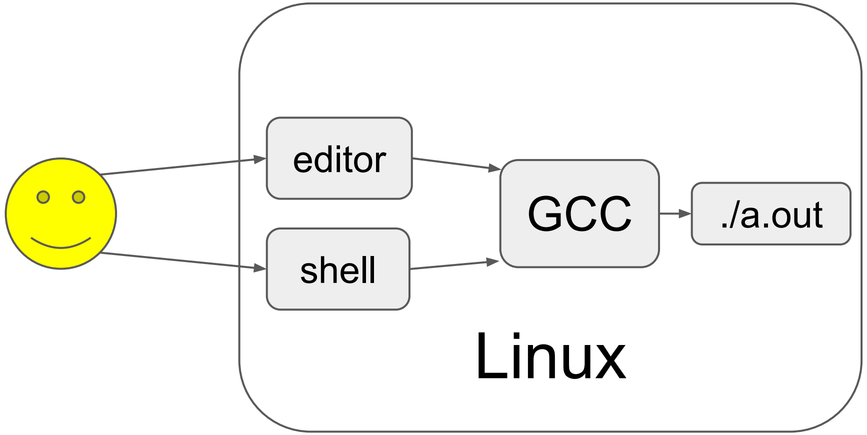


Magic == tools

   Systems are complex
+ Complexity requires tools

You need to learn some tools!

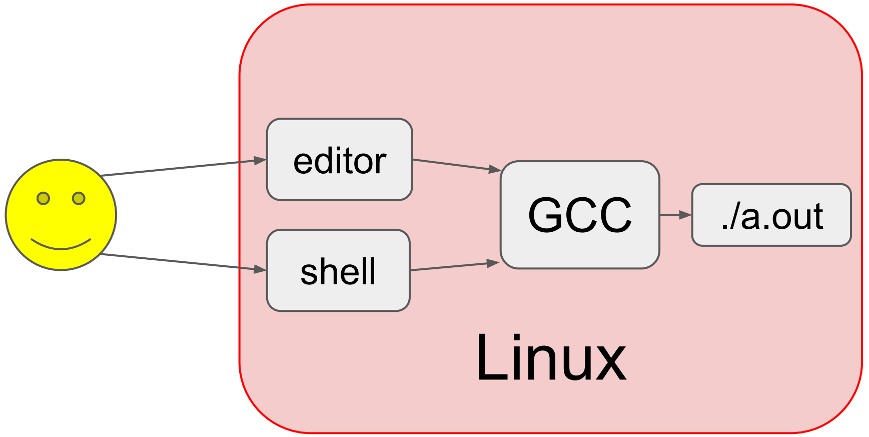
Lab 0 is about getting comfortable



CSE 391 Unix Tools, 1 credit

Linux

- You need this to run any of the tools
- Centos VM
- ssh into attu (if CSE)

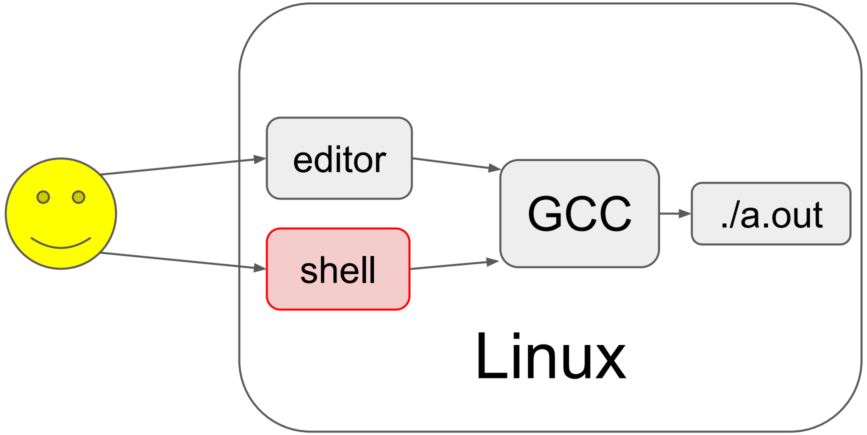


You should have one of these working,

ask for help if not!

Shell

- How you use
- Course page tutorial
- `man`
  - `man 3`
- Worth checking out the 391 website even if you're not in the class (cs.uw.edu/391)

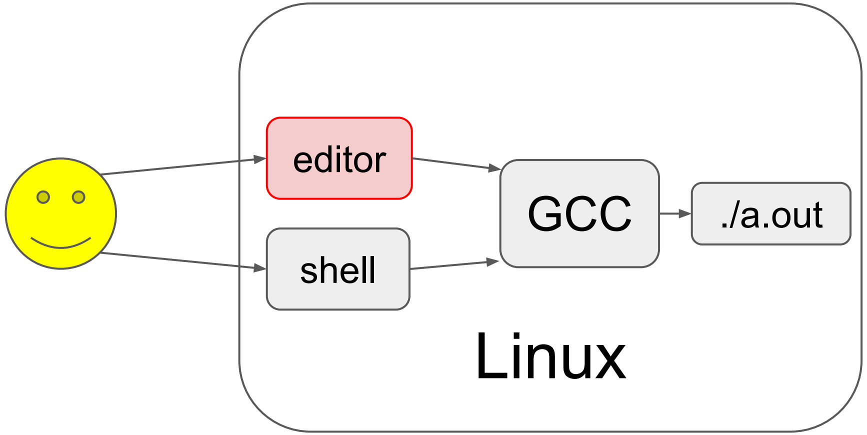


```
ls cd mkdir cp mv rm ...
```

You'll need to know these commands

Editor

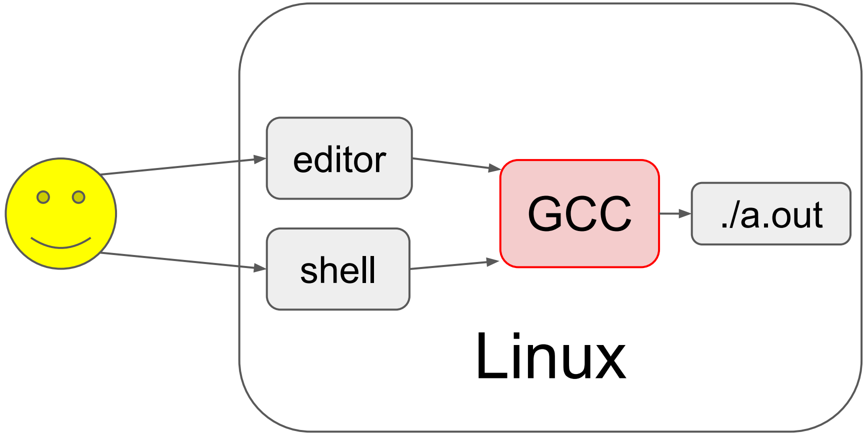
- Personal preference
- With great power, comes great responsibility (and learning)



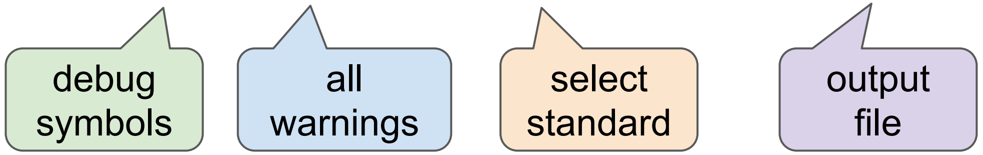
	Simple	Powerful
Graphical	Gedit	Emacs
Terminal	Nano	Vim

Compiler

- We'll use GCC (there are others)
- Lots of options (`man gcc`)
    - You only need a few



```
gcc -g -Wall -std=gnu99 -o arrays arrays.c
```



Hello world

hello.c

```
gcc hello.c -o hello
```

C Preprocessor
Standard Input/Output

Arguments from command line
Not important for now

Start here

```c
#include <stdio.h>

int main (int argc, char* argv[]) {
  // Declare then assign
  int x;
  x = 2;

  // Or do both
  int y = 5;

  // Print a formatted string
  // Note that \n is a newline
  printf("Hello world!\nx + y = %d\n", x + y);

  // Note the return type of main is int
  // A program typically returns 0 if everything went ok
  return 0;
}
```

Format specifier
Look here or `man 3 printf`

Escape sequence

Declared in `stdio.h`

Calculator

A little bit more substantial

calculator.c

```
gcc calculator.c -o calc
```

```
./calc 2 2 +
```

Try to add support for division (watch out for zero!)

More resources

These are on the schedule too!

C Cheatsheet

Emacs Cheatsheet

Unix shell Cheatsheet