Basic Linux Skills

oh, cool

who am i?

- i’m cyle
- i’m a systems developer and architect
- i use linux all day every day
- i like this kind of stuff

why use linux?

- specifically, a no-GUI linux
- it's fast. so fast.
- resources go to services, not the OS
- unix conventions > windows conventions

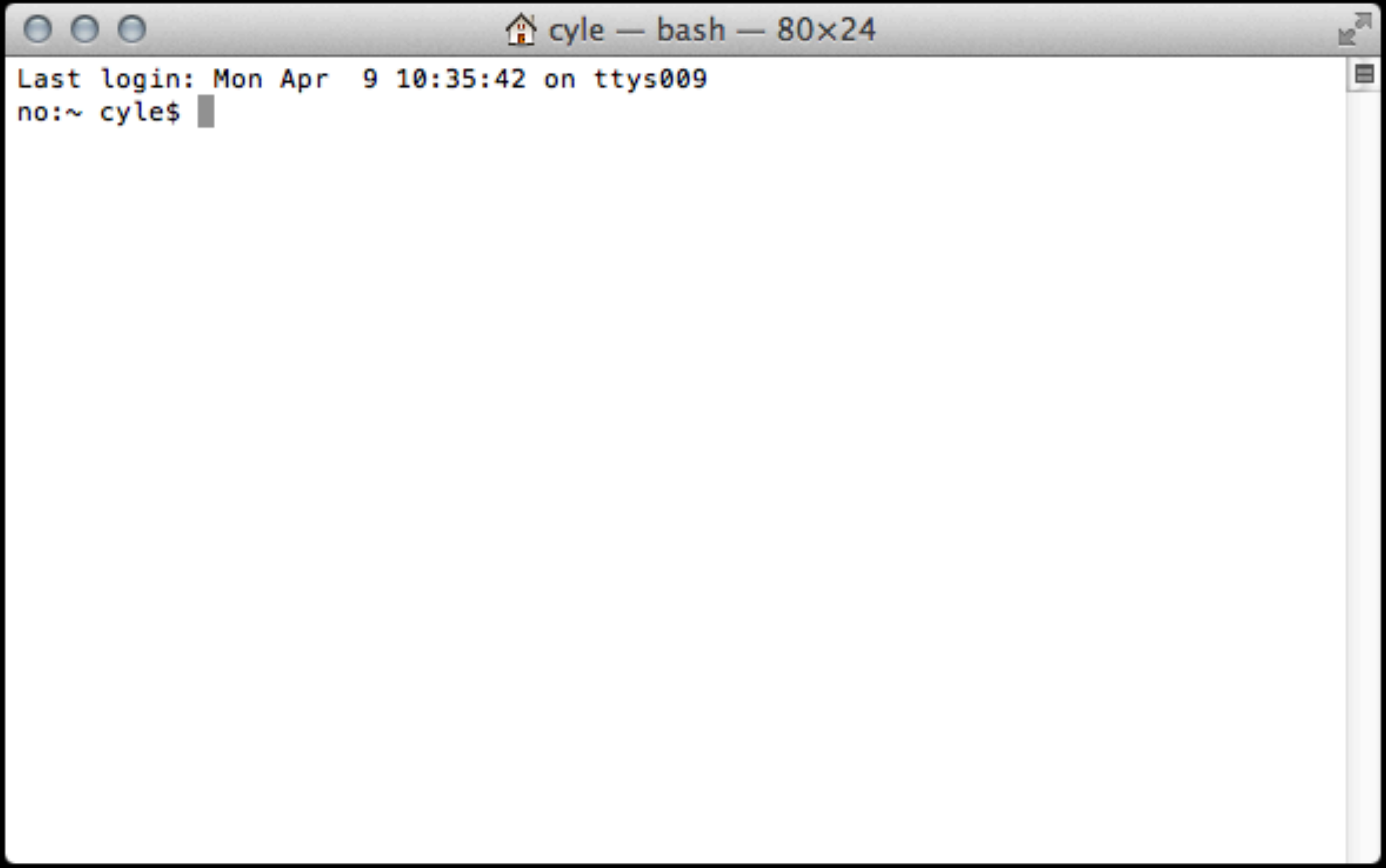
linux distributions

- “flavors”
- same idea, different implementations
- super free, super open-source
- Debian (highest used server distro)
    - Ubuntu, Mint
- Red Hat
- and so many more...

it’s not magic

- it’s just a computer, like any other
    - runs programs, etc
- you just type stuff instead of using a mouse

this is where you live



```
Last login: Mon Apr  9 10:35:42 on ttys009
no:~ cyle$ 
```

(or PuTTY if you’re on a PC)

connect remotely

- the majority of the time, you connect to a server remotely
- the server will be off in a datacenter somewhere
- `ssh user@server.com`

open Terminal

```
ssh user@learn.dev.emerson.edu
```

life in the CLI

- you’re in a “shell” right now
- it’s a lot like an iPhone (before multitasking)
- one thing running at a time
- you live in the file system
  - until you run a program

user@learn:~$

- WHAT IS THAT?
- it's called a "prompt"
  - each distro looks a little different, but there's a common theme

user@learn:~$

- that’s you! or rather, who you’re logged in as

user@learn:~$

- that’s an @ symbol... like an email address

user@learn:~$

- that’s what machine you’re on
- it’s the hostname (as it sees itself)

user@learn:~$

- that’s just a colon... acts as a divider

user@learn:~$

- that's where you are right now
- the ~ means "your home directory"
- if you type `pwd`, it'll return...
- /home/user

user@learn:~$

- signifies you are a normal user
- it would be a # if you were root

in the shell

- you execute commands by typing them in
- commands are just words you type
    - next slide will have examples
- in a GUI, you're doing this without knowing it

moving around

- `ls` (lets you see the contents of the active directory)
- `cd` (lets you change directories)
- `pwd` (shows you where you are)
- /, ./, and ../

arguments

- echo hello
- echo “hello there”
- cd ../
- nano what.txt
- less what.txt
- “what.txt” is an argument, or parameter

command switches

- ls -lah
- ls -l -a -h

- the -a is a “switch” for the command
  - it means show me ALL the files, even hidden ones
- switches activate additional functionality for the command
- almost every command has potential switches

common commands

- mv, cp, rm, mkdir, rmdir
- exit
- shutdown -h now
  - that’s a switch AND an argument
- reboot

some neat shell tips

- TAB
- double-TAB
- press UP
- lots more

man pages

- /usr/bin, /bin, /usr/sbin, etc
- man less
- 99% of commands have a “man” page

important note about users on linux

- like on windows or mac, users have their own “home folders”, have their own “groups”
- except one user... root. it is all-powerful.
- in normal systems administration, you’d log in as root, and do whatever you want
- but normal users are limited in what they can do; but normally they can use sudo

sudo

- prefixing your shell command with “sudo” lets you run that command as root, bypassing any security considerations
- … if the server admin lets you
- example:
- `nano /etc/hostname` (that won’t work)
- `sudo nano /etc/hostname` (that might)

useful commands

- top
- chmod / chown
- wget
- nano
- ping
- find

places to be in linux

- /var/log
- /etc
- /usr/src
- /var/www
- ~

logs are great

- pretty much everything in linux is logged
- it’s awesome, so useful
- even what you’re typing is logged!

services

- stuff running in the background at all times
- try running top or htop
- those are things running right now!
  - like Activity Monitor on Mac or Task Manager on Windows
- some of them are services, some might be other users on the system

services, cont'd

- services run in the background
  - also known as daemons
- you can use the `service` command to act on them
- for example, web server acting up?

  -
    ```
    service apache2 restart
    ```
- see all services:

  -
    ```
    service --status-all
    ```
- (you probably need to use sudo to do this stuff)

installing stuff

- you can install stuff in two ways:
  - packages (easy, awesome)
  - from source (complicated)
- on debian-based systems, `apt-get`
- `apt-cache` or `aptitude` for looking for stuff
- however, this is *system-wide* installation

compiling programs from source

- is not something i’m going to cover here
- but it’s fairly straightforward *if the developer has provided documentation*
- otherwise, good luck
- node.js is a good example of good docs
  - and all the cool kids are using it

build a web server

- it’s literally this easy to build a LAMP stack...
- `apt-get install apache2 libapache2-mod-php mysql-server`
- now you have linux, apache, mysql, and php
- put stuff in /var/www to use it
- restart the apache service whenever you edit apache or php’s configuration files

that’s...

- pretty much it.
- yeah, really.
- get your brain used to a CLI
- build your own server using VirtualBox

more learning

- http://arepository.com/of/learning/
- email me
  - cyle_gage@emerson.edu