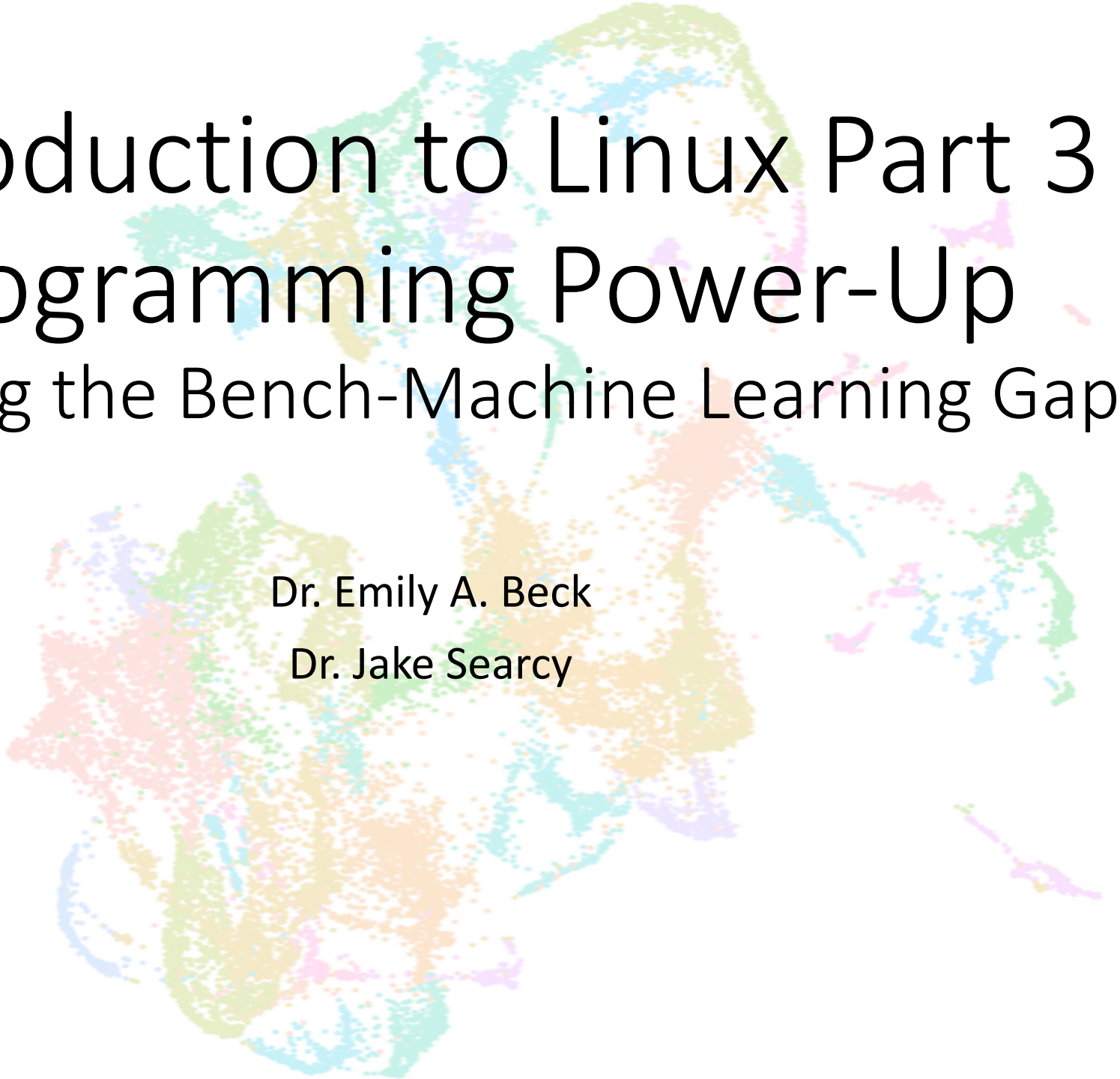
Introduction to Linux Part 3 Programming Power-Up

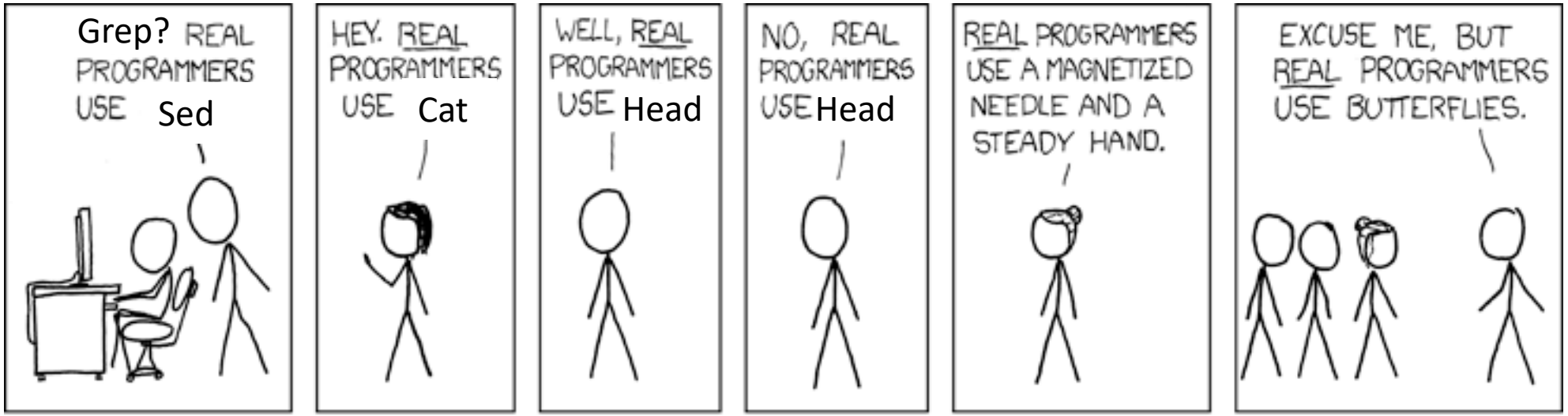
Bridging the Bench-Machine Learning Gap

Dr. Emily A. Beck

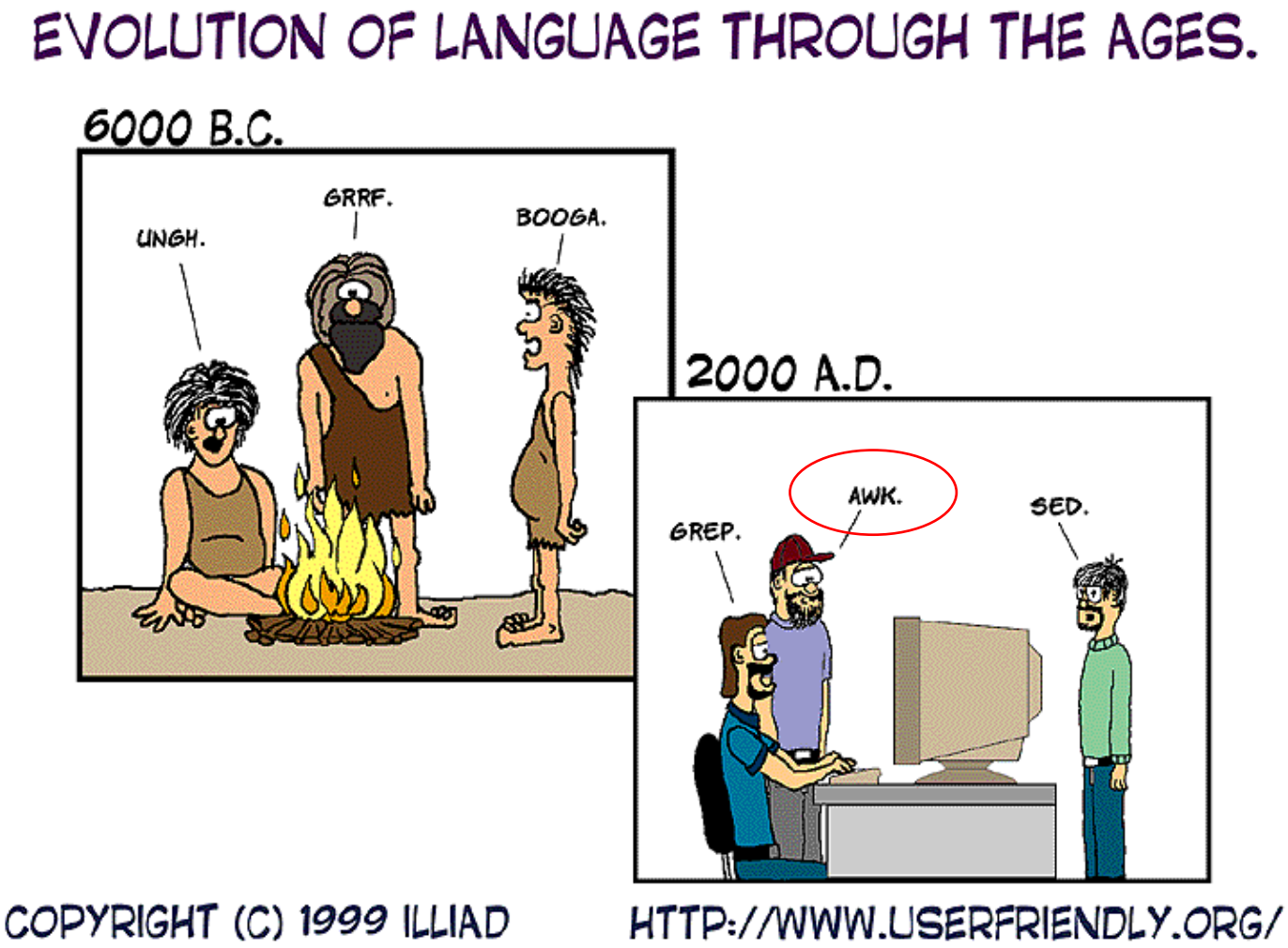
Dr. Jake Searcy



There are MANY ways to do a task and MANY tools to help you! Use what works for YOU.



Grep/Sed/Awk are three powerful tools to accomplish most common text manipulations in Linux



Learning Objectives

- Become comfortable with Awk for data frame manipulation
- Learn useful tips and tricks for programming more efficiently
  - How to parallelize programs to be more efficient
  - How to use pipes to streamline your code

Awk: an excellent extraction tool

awk can be used for searches, printing, and manipulations commonly needed when looking at large files. Can commonly replace grep and sed.

Similar to grep but with additional built in additional functions
and a slightly different syntax

‘{print(x)}’- print a field
‘{length(x)}’- print the length of a field
‘{sub(/x/,y)}’- substitute y for x (first occurrence)
‘{gsub(/x/,y}’ – global substitution y for x
Also built in mathematical functions:
‘{sqrt(x)}’ – calculate the square root of X

Awk is great for working with data frames

Example:

If I have a large tsv file and I only want columns 1, 3, and 7 I can use awk

awk ‘{print $1, $3, $7}’ large_file.tsv > subset.tsv

If we are working with other file types (for example csv) we need to note what the field separator is:
–F fs where fs is the field separator

Csv example:
–F ,

Sort is another useful command

sort <file.txt>
-k indicate specific column
-t fs indicate a field separator
-r sort in reverse alphabetical order
-n sort numerically

Pipes: Useful when performing multiple steps

Sometimes you want the output of one process to be the input of another

You could run program 1 , wait and run program 2.
OR you could use a *Pipe* using | vertical line on your keyboard

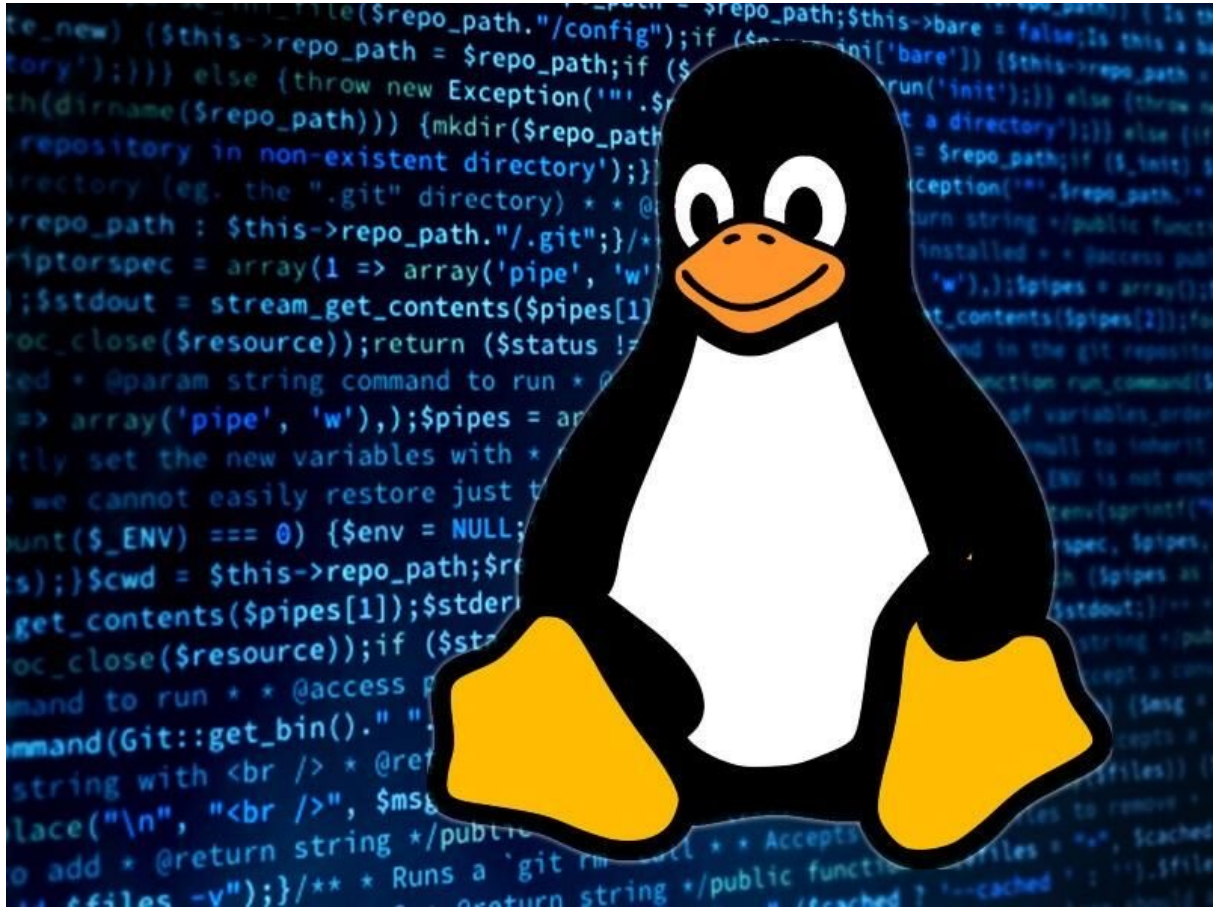
Example:
I once again have a large file *addresses.csv*
Name, address, city, state, phone number

I want a new file sorted alphabetically by State with just names, city, state.

cat addresses.csv|sort –t, -k4|awk –F, '{print $1,$3,$4} > subset.csv

Let’s play around with pipes

Think of some fun manipulations for the penguin.csv file and try to make your own pipe



Awk: an excellent extraction tool

awk can be used for searches, printing, and manipulations commonly needed when looking at large files. Can commonly replace grep and sed.

Similar to grep but with additional built in additional functions
and a slightly different syntax

‘{print(x)}’- print a field
‘{length(x)}’- print the length of a field
‘{sub(/x/,y)}’- substitute y for x (first occurrence)
‘{gsub(/x/,y}’ – global substitution y for x
Also built in mathematical functions:
‘{sqrt(x)}’ – calculate the square root of X

Screens: Useful when running long programs

Sometimes things take a while to run especially when working with large files

Maybe you want to run a program on a lot of files at the same time?

You could open a bunch of terminal windows all at once and have them run at the same time or you could set a screen!

Screens allow for programs to run in the background. You can detach from screens to let programs run and reattach to check on how things are going

How to set, detach, reattach, and kill screens

To view a list of existing screens:
screen –ls

To set a new screen:
screen –S name

*Try to get in the habit of naming your screens something useful. You will thank yourself later when you have 10 things going at once and you want to check in on the right screen without having to check all ten

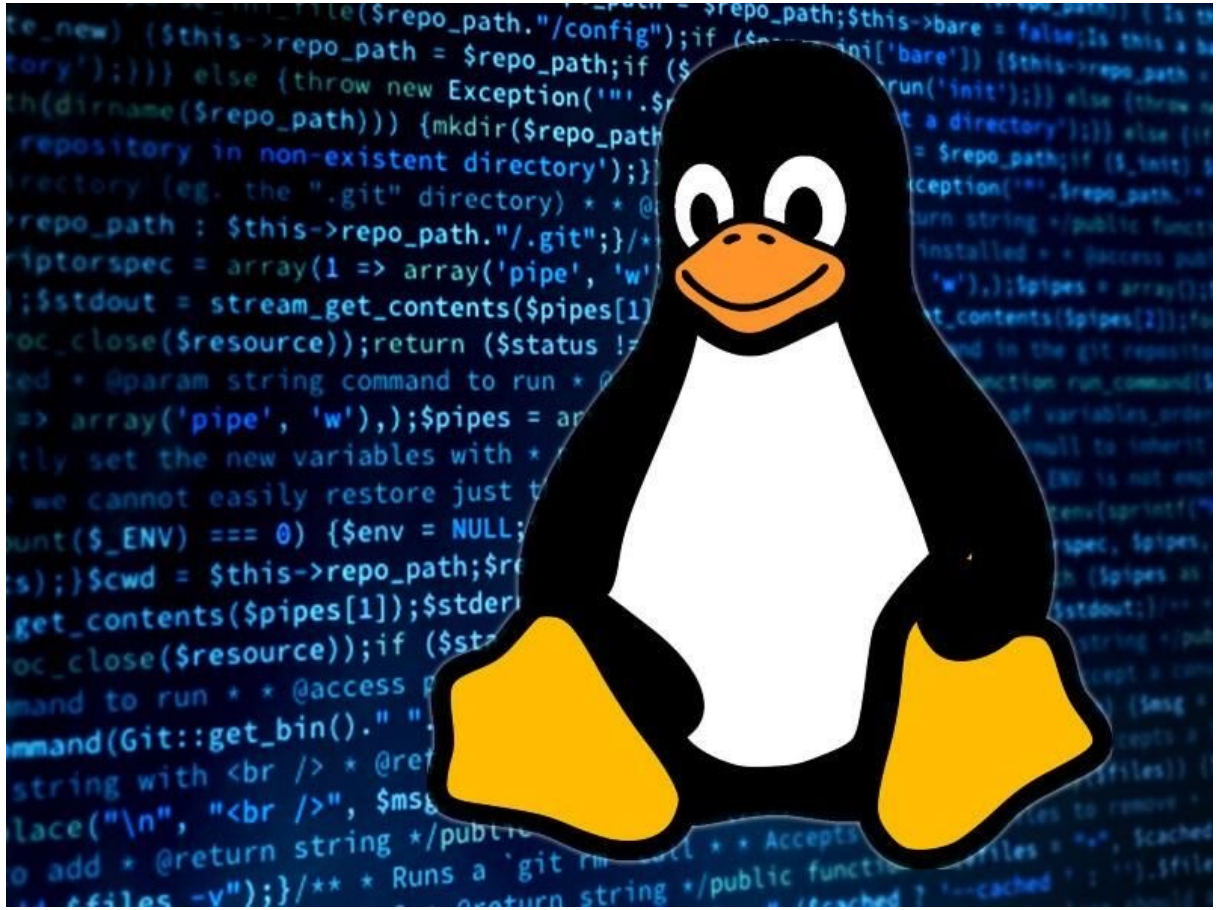
To detach from a screen:
Crtl a+d

To reattach to a screen:
screen -r name

To kill a screen: (When you are done and no longer want to use the screen)
screen kill –S –X name

Let’s play around with screens!

Make some screens, attach, reattach, kill them etc.



How to set, detach, reattach, and kill screens

To view a list of existing screens:
screen –ls

To set a new screen:
screen –S name

*Try to get in the habit of naming your screens something useful. You will thank yourself later when you have 10 things going at once and you want to check in on the right screen without having to check all ten

To detach from a screen:
Crtl a+d

To reattach to a screen:
screen -r name

To kill a screen: (When you are done and no longer want to use the screen)
screen kill –S –X name


BREAK TIME