Bash Tutorial Summary

COMP 206
Fall 2017

Edit from command line

```
[1]$ vi script.sh
```

use vi, or any other text editor you like & create a text file and name it script.sh

Write a script that:

1. Checks the current day of the week
2. If it's Friday, display
   "Thank goodness it is Friday!"
3. If it's the weekend, display
   "You should not be working, go home"
4. For all other weekdays, display
   "Time to work."

Simple Bash Script to Run

```
#!/bin/bash
set `date`
if test $1 = Fri
then
  echo "Thank goodness it is Friday!"
elif test $1 = Sat || test $1 = Sun
then
  echo "You should not be working"
  echo "Go home"
else
  echo "Time to work."
fi
```

MAKE SURE:
You're using backticks: `
and not apostrophes: '
THEY ARE DIFFERENT

Run from command line

```
[1]$ vi script.sh
[2]$ chmod 755 script.sh
```

Run from command line

```
[1]$ vi script.sh
[2]$ chmod 755 script.sh
[3]$ ./script.sh
```

Run from command line

```
[1]$ vi script.sh
[2]$ chmod 755 script.sh
[3]$ ./script.sh
```

What’s the result?

Now add to the script:

1. Get input from user: ask for their name

2. Print their name alongside the “echo” statements.

How do you take in $arguments from the command line?

```
#!/bin/bash
set `date`

echo "Hello, what is your name?"
read name
```

Use read for input
&
$ to access arg

```
if test $1 = Fri
then
  echo "Hey $name , Thank goodness it is Friday!"

elif test $1 = Sat || test $1 = Sun
then
  echo "Hey $name, You should not be working"
  echo "Go home"
else
  echo "Hey $name, Time to work."

fi
```

How do you take in $arguments from the command line?

```
#!/bin/bash
set `date`

echo "Hello, what is your name?"
read name

if test $1 = Fri
then
  echo "Hey $name, Thank goodness it is Friday!"

elif test $1 = Sat || test $1 = Sun
then
  echo "Hey $name, You should not be working"
  echo "Go home"
else
  echo "Hey $name, Time to work."

fi
```

SOLVE THIS PROBLEM

We want to display all the **users currently logged into the computer** when it is **time to work**.

Then use `grep` to search the names of all the users that are logged in, to see if your friend is present.

Display your output like this:

`ALL LOGGED IN USERS` [displays all the users].
`MY FRIEND` [display your friend || nothing if they are not present].

HINT: **Store your list in a file, then access the file!**

SOLUTION

```
#!/bin/bash
set `date`

echo "Hello, what is your name?"
read name
if test $1 = Fri
then
  echo "Hey $name, Thank goodness it is Friday!"

  echo "ALL LOGGED IN USERS"
  who  >> users.txt
  cat users.txt
  echo "MY FRIEND:"
  grep “zhang" users.txt

elif test $1 = Sat || test $1 = Sun
then
  echo "Hey $name, You should not be working"
  echo "Go home”
else
  echo "Hey $name, Time to work."
fi
```

This is how our code looked after previous exercise

SOLUTION

```
#!/bin/bash
set `date`

echo "Hello, what is your name?"
read name
if test $1 = Fri
then
  echo "Hey $name, Thank goodness it is Friday!"

  echo "ALL LOGGED IN USERS"
  who  >> users.txt
  cat users.txt
  echo "MY FRIEND:"
  grep “zhang" users.txt

elif test $1 = Sat || test $1 = Sun
then
  echo "Hey $name, You should not be working"
  echo "Go home"
else
  echo "Hey $name, Time to work."
fi
```

Use who to look for online users

Use grep to search the file of users

SOLUTION

```
#!/bin/bash
set `date`

echo "Hello, what is your name?"
read name
if test $1 = Fri
then
  echo "Hey $name, Thank goodness it is Friday!"
  echo "ALL LOGGED IN USERS"
  who  >> users.txt
  cat users.txt
  echo "MY FRIEND:"
  grep “bob" users.txt

elif test $1 = Sat || test $1 = Sun
then
  echo "Hey $name, You should not be working"
  echo "Go home"
else
  echo "Hey $name, Time to work."
fi
```

Save to a file so it’s easy to
display (cat)
search & display if found (grep)