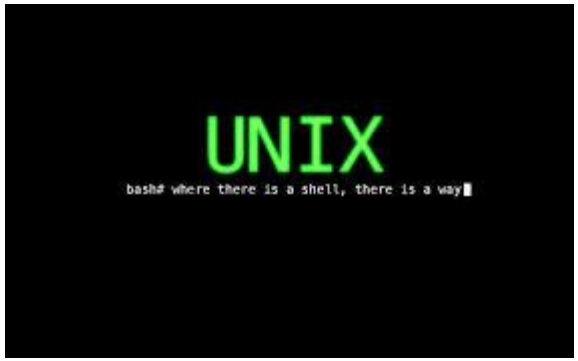
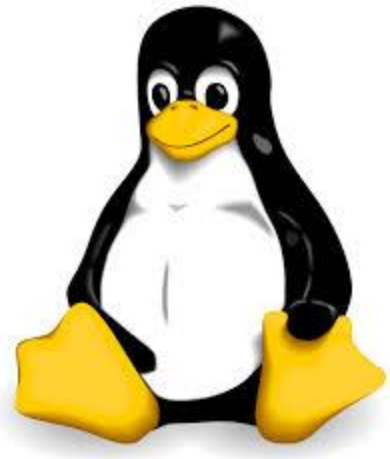
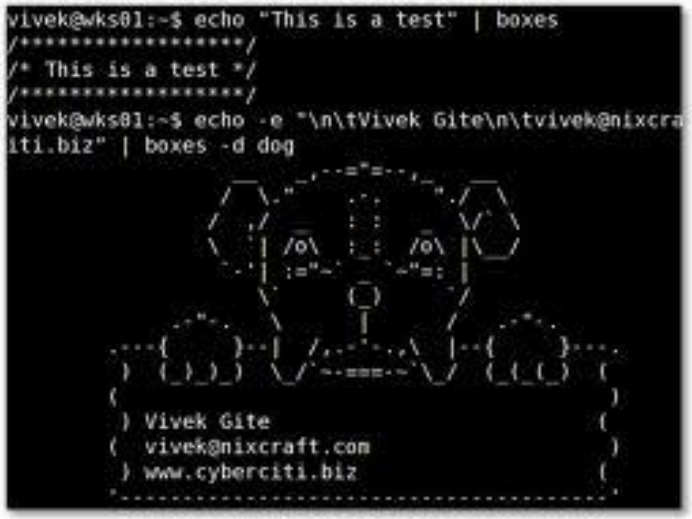
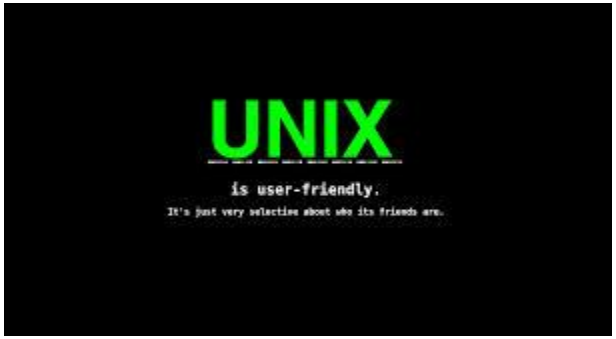
Introduction to Scientific Computing
Meeting 8
Unix Commands









Jeremiah Lant, Hydrologist
USGS Kentucky Water Science Center
jlant@usgs.gov

Review

- Find the largest (peak) discharge value in May of 2013
- **KEY:** build incrementally and test

```
$ grep 2013-05-* 03290500_dv.txt | head
$ grep 2013-05-* 03290500_dv.txt | cut -f 4 | head
$ grep 2013-05-* 03290500_dv.txt | cut -f 4 | sort -n | head
$ grep 2013-05-* 03290500_dv.txt | cut -f 4 | sort -n | uniq | head
$ grep 2013-05-* 03290500_dv.txt | cut -f 4 | sort -n | uniq | tail
$ grep 2013-05-* 03290500_dv.txt | cut -f 4 | sort -n | uniq | tail -1
```

- If you wanted to know the date as well as the value

```
$ grep 2013-05-* 03290500_dv.txt | cut -f 3-4 | head
$ grep 2013-05-* 03290500_dv.txt | cut -f 3-4 | sort –n –k2  | head
$ grep 2013-05-* 03290500_dv.txt | cut -f 3-4 | sort –n –k2  | uniq | head
$ grep 2013-05-* 03290500_dv.txt | cut -f 3-4 | sort –n –k2  | uniq | tail
$ grep 2013-05-* 03290500_dv.txt | cut -f 3-4 | sort –n –k2 | uniq | tail -1
```

Review

- Find the largest (peak) discharge value in entire file and the date
- **KEY:** build incrementally and test
- **Hint: -v means to invert match**
- **Hint: -e means use what follows as the pattern**

$ grep -v -e “#” 03290500_dv.txt | head

$ grep -v -e “#” -e “agency_cd” 03290500_dv.txt | head

$ grep -v -e “#” -e “agency_cd” -e “5s” 03290500_dv.txt | head

$ grep -v -e “#” -e “agency_cd” -e “5s” 03290500_dv.txt | cut -f 3-4 | head

$ grep -v -e “#” -e “agency_cd” -e “5s” 03290500_dv.txt | cut -f 3-4 | sort –n –k2| head

$ grep -v -e “#” -e “agency_cd” -e “5s” 03290500_dv.txt | cut -f 3-4 | sort –n –k2| uniq | head

$ grep -v -e “#” -e “agency_cd” -e “5s” 03290500_dv.txt | cut -f 3-4 | sort –n –k2| uniq | tail

$ grep -v -e “#” -e “agency_cd” -e “5s” 03290500_dv.txt | cut -f 3-4 | sort –n –k2| uniq | tail -1

Review

- **More hands-on learning** with Unix commands
  - Used **cut** to cut out columns from a data file
  - E.g. stage-csv.txt and stage-tsv.txt

```
Date,Parameter,Value
2014-04-29,stage,17
2014-04-30,stage,20
```

Date	Parameter	Value
2014-04-29	stage	17
2014-04-30	stage	20

$ cut -d , -f 1 stage-csv.txt **# cuts by comma delimiter and gets column (field -f) 1**

$ cut -d , -f 1-3 stage-csv.txt **# cuts by comma delimiter and gets columns (field -f) 1-3**

$ cut -f 1 stage-tsv.txt **# cuts by tab delimiter and gets column (field -f) 1**

$ cut -f 1-3 stage-tsv.txt **# cuts by tab delimiter and gets columns (field -f) 1-3**

Today's Objective

1. **Make and run a bash script; *.sh file**
2. **Introduce the for loop;** A loop that is executed once for each value in some kind of set, list, or range.

- **Why?**
  - Good basis for learning how to program.
  - Becoming more comfortable on command line
  - Becoming more efficient

Make a bash script

- Redirect the last command to a file called peak.sh
- **Hint:** use history command to help

$ history | tail -5 > peak.sh

$ cat peak.sh

$ start notepad++ peak.sh

$ bash peak.sh

For Loop

- Find the largest value and date for multiple daily value files?
- **for loop** - A loop that is executed once for each value in some kind of set, list, or range.

- Create a for loop to add “processed” as a prefix to a set of files
- Make a temporary directory and create 5 “dummy” files called data1.txt, data2.txt, data3.txt, data4.txt, data5.txt

```
$ for filename in *.txt
> do
>         echo $filename
> done
```

- Note, the above is the same as the following:

```
$ for filename in *.txt; do echo $filename; done
```

For Loop

- Find the largest value and date for multiple daily value files?
- **for loop** - A loop that is executed once for each value in some kind of set, list, or range.

- Create a for loop to add “processed” as a prefix to a set of files
- Make a temporary directory and create 5 “dummy” files called data1.txt, data2.txt, data3.txt, data4.txt, data5.txt

```
$ for filename in *.txt
> do
>         echo $filename
>         mv $filename processed-$filename
> done
```

- Note, the above is the same as the following:

```
$ for filename in *.txt; do mv $filename process-$filename; done
```

For Loop

- What if you wanted to know the largest value for multiple daily value files?
- Copy the peak.sh as peak2.sh
- Hint: peak.sh only works for daily value files

```
for filename in *dv.txt
do
        echo $filename
done
```

```
for filename in *dv.txt
do
        echo $filename
        grep -v -e “#” -e “agency_cd” -e “5s” $filename | cut -f 4 | sort - n | uniq | tail -1
done
```

Next meeting

- Version control with Git