

# BASH Files and Directories

---

SCRIPTING ESSENTIALS

DR. BURKMAN

# Helpful student suggestion

---

Caleb Russell gives this good advice:

- I changed a setting for that, so I can use our openSUSE in fullscreen with a nicely stretched image. It was at the top-left of the VMWare window, the VMRC button, Preferences; there's an option in the middle of that Preferences window for VM Window Resize to "Stretch the Virtual Machine in the window."

Thanks Caleb!

# A note on Cut

---

To make a string to an array you must cut on a space!

Mod 07 Slide 5 shows cutting a string to make an array with delimiters but you can also cut strings by delimiters to make or change strings.

- `my_string=$(echo $my_string | cut -d "," -f 1)` for example

```
# my_string="dog,cat,mouse"
# my_string=$(echo $my_string | cut -d "," -f 1)
# echo $my_string

# my_string="rat,bat,wren"
# my_array=($(echo $my_string | cut -d "," -f 1-))
# echo ${my_array[0]}

# my_string="rat,bat,wren"
# my_string=$(echo $my_string | tr "," " ")
# my_array=($(echo $my_string | cut -d " " -f 1-))
# echo ${my_array[0]}
```

# Basic Linux Commands

---

`rm`

- Removes a file

`rm -r`

- Removes a directory and everything in it

`mkdir`

- Creates a directory

To make a file readable:

- `chmod 640 <filename>`

# “Zipping” in Linux

---

## tar

- <https://www.gnu.org/software/tar/manual/tar.html>
- <http://www.linfo.org/tar.html>
- For now:
  - `tar -cf <archive name> <file1> <file2>` (makes archive)
  - `tar -cf <archive name> <file1> <file2> -z` (makes archive and compresses)
  - `tar -xf filename.tgz` (unpacks an uncompressed archive)
  - `tar -xzf filename.tgz` (unpacks and decompresses)
  - `tar -xzf filename.tgz -C /target/directory` (puts in another directory)

# Reading a File

---

Do not put quotes around path for input!!!!!!

```
input=alert.csv
while IFS= read -r line
do
    echo $line
done < $input
```

Here I'm also outputting to a file. > is destructive. >> appends.

# Writing to a File

---

> will delete the file first if it exists

>> will append to an existing file or make the file if it doesn't exist

```
#destructive write
my_output="this,that,more"
echo $my_output > m08x.csv

one="one"
two="two"
three="three"

#append write_
my_output="$one,$two,$three"
echo $my output >> m08x.csv
```

# Checking for a file or directory

---

There is nothing special about the word FILE. Use -d for directory check

```
FILE=/etc/resolv.conf
if [[ -f "$FILE" ]]; then
    echo "$FILE exists."
fi
```



# Writing long lines

---

Writing out and line length errors. Cannot have extra white space after the \

```
array_len=${#employee_array[*]}
for (( i=0; i<$array_len; i=$((i + 3)) ))
do
    #be sure to write out like this so no line length errors
    line_out="${employee_array[$i]},\
${employee_array[$(($i+1))]},\
${employee_array[$(($i+2))]}"
    echo -e $line_out >> ~/mod08t/employee.csv
done
```

# Tutorial

---

This will all be expanded on in the tutorial. I highly recommend taking your time with it, and making sure that you understand everything. The project is next week!