

HW 1

Mike Hanling

17 JAN 2018

1. (10 points) What are the three components of a Unix system? Provide an example of the interaction from a user perspective down to the hardware and back up.

User Space, Kernel Space, Hardware. Ex: The user can use the `whoami` command in the Shell. The kernel will then interpret the command to find a certain memory address that holds the user's name. The kernel will have to tell the hardware to do this lookup. The hardware will do this and send back the answer (in binary of course) to the kernel. The kernel can then interpret the machine language to produce the user's name back.

2. (15 points) Consider the `ls -l` output below, label the output appropriately:

```
drwxr-xr-x 2 aviv scs 4096 2013-12-22 10:57 demo/  
-rw-r--r-- 1 aviv scs 13454 2013-12-22 10:56 text.dat
```

The first line of output:

Directory

Read permissions for user, group, and global

Write permissions for user only

Execute permissions for user, group, and global

Owned by aviv

In group scs

4096 bits in size

Last modified 12DEC2013 at 1057

Named demo

The second line of output:

File

Read permissions for user, group, and global

Write permissions for user only

No execute permissions

Owned by aviv

In group scs

13454 bits in size

Last modified 12DEC2013 at 1056

Named text.dat

3. (15 points) For the following commands, determine in which bin directory they live by using which command on a lab machine

(a) `ls`

`/bin/ls`

(b) `which`

`/usr/bin/which`

(c) `tac`

`/usr/bin/tac`

(d) `grep`

`/bin/grep`

(e) `cut`

`/usr/bin/cut`

(f) `chmod`

`/bin/chmod`

(g) `head`

`/usr/bin/head`

(h) `mv`

`/bin/mv`

4. (10 points) Look up the `tac` command in the man pages. Describe its operations, and give an example usage.

```
tac concatenates and prints files in reverse. It does the reverse of cat. Ex:
cat nums.txt
1
2
3
tac nums.txt
3
2
1
```

5. (15 points) What are the three guiding principles of the Unix design philosophy?

- Write programs to do *one thing* & do it well.
- Write programs to *work together*.
- Write programs to handle *text streams*, because that is a universal interface.

6. (10 points) What are the primary purposes of standard input, output, and error for different programs?

<code>stdin</code>	Input stream for reading anything typed in the terminal
<code>stdout</code>	Output stream for printing anything to the terminal
<code>stderr</code>	Separate output stream for errors in programs (not program output)

7. (15 points) Consider the following command line with redirects:

```
grep PA < sample-db.csv 2> oops > sample-db.PA.csv
```

(a) What is the output file?

sample-db.PA.scv

(b) What is the input file?

sample.db.scv

(c) What is the error file?

oops

8. (10 points) Using a pipeline as an example, why is it necessary to have standard error and standard output?

`tac FILE_YOU_DO_NOT_HAVE_READ_PERMISSIONS_FOR | cat something.txt - else.txt`
Since you do not have the permissions to perform `tac`, there will be an error message. In order to not have `cat` print out the error message in the middle of its output to standard output, there must be a standard error stream to keep error messages and program output separate.