

## Administrivia

- Homework 3 (plus reading) coming soon.

Slide 1

## Shell Customizations

- At startup, shell reads in various configuration files (see man page for details). At least one will be in your home directory (`.bashrc` for `bash`).
- In this file, you can
  - Define/redefine environment variables (e.g., `PATH`, `PS1`). For `bash`, be sure to `export` them. Can define new ones (I find this useful).
  - Define aliases/functions. For `bash`:
    - \* Aliases are simple substitution, no parameters. E.g.

```
alias lt='ls -ltF'
```

```
alias google='lynx http://www.google.com'
```
    - \* Functions can have positional parameters. E.g.,

```
function cd-and-show() { cd $1 ; pwd ; ls; }
```

Slide 2

### I/O Redirection, Recap

- Standard input can come from keyboard, file (`< file`, "here document" (more later — useful in scripts), pipe.
- Standard output can go to keyboard, file (`> file`, `>> file`), pipe. Same for standard error. Syntax details vary from shell to shell.

Slide 3

### Pipes, Review

- "Pipes" provide one-way communication between programs — output of program A becomes input of program B.
- Key component of "the Unix philosophy" — emphasis on providing a toolkit of small programs, mechanisms for combining them.
- "Filters" are programs designed to work this way — text input, text output, simple regular format.

Slide 4

## Filters

- Some commonly-used filters:

```
head  tail
sort  uniq
grep  wc
cut   paste
tr    expand
awk   sed
```

- Use these in combination with, e.g., `ps`, `ls`.

Slide 5

## Examples

- Find all processes that belong to your username.

```
(ps aux | grep $USER)
```

- Find all users who are running processes on the system.

```
(ps aux | awk '{ print $1 }' | sort | uniq)
```

- Generate a list of machines that are "up".

```
(ruptime | grep up | awk '{ print $1 }')
```

- Show how much space each subdirectory of your home directory is using, sorted by size.

```
(du -sk $HOME/* | sort -n)
```

Slide 6

### More Useful Commands

- `find`. Examples:
  - Find all files in the current directory created in the last week.  

```
(find . -mtime -7)
```
  - Find all files in your home directory whose name contains `hello`.  

```
(find $HOME -name "*hello*")
```
  - Find all files in the current directory that end in `.bak` and apply `rm -i` to them.  

```
(find . -name "*.bak" -exec rm -i {} \;)
```

Slide 7

### More Useful Commands

- `diff`.
- `xargs`. Example:
  - Find all processes for program `xcpustate` and kill them:  

```
(ps aux | grep xcpustate | awk '{ print $2 }'  
| xargs kill)
```

Slide 8

### Minute Essay

- What command could you use to find all aliases defined in your `.bashrc` file and print them out in sorted order?
- Reminder: Homeworks 1 and 2 due by 5pm.

Slide 9