

Slide 1

Administrivia

- Reminder: Homework 1 due today at 5pm.
- Homework 2 on Web soon; due next Monday.

Slide 2

Shell Features — Recap

- Things that happen while entering command — editing, tab completion, use of command history.
- Parsing of completed command line — filename wildcard expansion, environment variables, separation into arguments, single and double quoting.
- Aliases and functions.
- How to save customizations for next login.

I/O Redirection

- `stdin` can come from keyboard, file, or "inline". (Review syntax.)
- `stdout`, `stderr` can go to terminal or file (overwrite or append), separately or together. (Review syntax.)
- *Or ...*

Slide 3

Pipes

- "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, and there are lots of them (some in next slides). `less` and `more` also useful.

Slide 4

Filters

Slide 5

- `head`, `tail`.
- `sort`, `uniq`.
- `grep` — search for text (or regular expression — more later).
- `wc` — count characters, words, lines.
- `tr` — “translate”. Good for converting, e.g., upper-case to lower-case.
- `tee` — duplicates input. Good for capturing output to a file while also displaying it onscreen.

Filters, Continued

Slide 6

- `sed` — “stream editor”. Example — convert DOS/Windows-style text file (each line ends with `\r\n`) to UNIX-style (each line ends with `\n`).
- `awk` — “pattern scanning and processing language” — many interesting possibilities; simplest is just to break up input into whitespace-delimited fields.

Examples

- Find all users on local machines and print their names only:

```
rwho | awk '{print $1}'
```

- Find all users on local machines and print their names only, suppressing duplicates:

```
rwho | awk '{print $1}' | uniq
```

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

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

Slide 7

Minute Essay

- What command could you use to count the number of aliases in your `.bashrc` file?
- Was Homework 1 too hard, too easy, or about right?

Slide 8

Minute Essay Answer

- One possible answer (to the first question):
`grep alias .bashrc | wc -l`

Slide 9