CS114: Lecture 5
Text processing, regular expressions

Have you started HW2?

Any questions on HW2?

Office hours tomorrow as usual

Small bug at the end of HW2 - “processes” should be “jobs”.

What's a text file?

- Contains only 7-bit ASCII text
- How do you tell if a file is a plain, ASCII text file?
- “Human readable”
- Not rich text – no **bold**, *italic*, FONTS, etc.
- Not Unicode
- “Universal” - can read on any operating system, with many programs
  - Except line breaks

Line breaks

- DOS/Windows world
  - Lines end in \r\n (0x13 0x10; ^M ^J)
- UNIX world
  - Lines end in \n

- Usually doesn't matter...
  - Notepad is stupid
  - Sometimes scripts won't run
  - Sometimes you'll see extra ^Ms

Where are text files used in UNIX?

- Everywhere!
- Configuration files
  - .cshrc, vimrc/.emacs, startup scripts
- Programs
  - Shell scripts, Perl / Python / Tcl / ..
- Data
  - Text documents (LaTeX)
  - “databases” (e.g. /etc/passwd)

Interactive searching in text files

- `vi, less -/`
- Emacs – Ctrl-S
- Nano – Ctrl-W

Programmatic searching: grep

- `grep` *`string file`*
  - Print lines of file containing string
- `grep -v` *`string file`*
  - Print lines of file not containing string
- What programs is Eric running?
  - `ps aux | grep ejb34`
- Where am I using the function drawline?
  - `grep drawline *.java`
- When have I used grep?
  - `history | grep grep`

Searching with regular expressions

- Related to wildcards, but more powerful – and different notation (grr...)
- `.` - any character
  - `grep a.b` finds lines containing `axb, ayb, a1b`, ...
- `^` - beginning of line
  - `grep ^1` finds lines beginning with `1`
- `$` - end of line
  - `grep 1$` finds lines ending with `1`

More regular expressions

- `[axb0-9]` – one character from this set (just like wildcards)
  - `grep ^[A-Z]` finds lines beginning with a capital letter
- *`expr`*`*` - any number of repetitions of *`expr`*
  - `grep ab*c` matches lines with ac, abbc, abbbbbc, ...

Extended regular expressions

- *expr*?
  - Match 0 or 1 `expr`s
- `expr+`
  - Match 1 or more `expr`s
- `expr{n,m}`
  - Match n, n+1, ..., m `expr`s
- `expr0|expr1`
  - Match `expr0` or `expr1`
- `(expr)`
  - Match `expr`; used for grouping

Special characters

- What if I want a real . * [^ $ + ? { ( ) |
  - Escape: \. * \[etc.
- BUT: Some programs require that you put a \ before some metacharacters: ? + { | (
  - “basic” grep, Emacs, vi, sed, ...
  - Here | matches |, but \| does “or”
- Some don't (“modern”):
  - Extended grep (`egrep` or `grep -E`), Perl
  - | does “or”, \| matches |

Search-and-replace: `tr`

- `tr` *`set1 set2`*
- Only stdin/stdout
- Use echo * but put newline between filenames
  - `echo * | tr ' ' '\n'`
- Make file all uppercase
  - `tr 'a-z' 'A-Z' < file`
- Delete everything but numbers
  - `tr -cd 0-9 < file`

`sed`, a stream editor

- `sed -e 's/`*`this`*`/`*`that`*`/' < file`
  - Replace *this* with *that; this* is a regular expression