Slide 1

Administrivia

- “Lecture topics and assignments” page will also have reading assignments, in online “Introduction to Linux” guide.
- Homework 1 on Web. Due next Monday. A key objective of this assignment is to get practice finding information in the `man` and `info` pages. I recommend working individually on this, at least at first.
- Homework 2 coming soon.

Slide 2

Minute Essay From Last Lecture

- Question: What are your goals for this course? Are there specific topics you’re interested in?

  Answers mentioned a range of topics, close to what I have in mind. Note that I don’t plan to address system administration.
- Question: Do you have access to a Linux or UNIX system other than the department’s lab machines?

  About half said “only the lab machines”. Should not be a problem.
  Off-campus access may be possible; come talk to me if you need/want this.

Slide 3

Files

- A key underlying concept — “everything’s a file” (sequence of bytes). Directories are files. Devices are represented as “special files”. Many files are text.
- Things to note:
  - Windows/DOS “extensions” idea doesn’t really apply.
  - Also no notion of “drive letters” — all paths form a single hierarchy. Removable media can be “mounted”.
  - Security model is simple but fairly flexible — rights (read, write, execute) for owner, group, others.
  - “Links” (hard or soft) allow non-tree directory structure.
- Be familiar with basic commands to manipulate/navigate filesystem.

Slide 4

Processes

- Another key concept — process as one of a set of “concurrently executing” entities (users, applications, etc.)
- Things to note:
  - Processes can spawn “child” processes.
  - Processes can have “environment variables”, inherited by child processes. Examples — `USER`, `PATH`.

Slide 5

Shells

- What's a "shell"? Program that interprets what you type, calls ("launches") other programs.
- This being Unix, there are several — `sh`, `bash`, `csh`, `tcsh`, `ksh`, `zsh`, etc.
- Most provide similar (and extensive) functionality, but syntax can vary. Read `man` page or manual for more info.
- Notice that some ctrl-whatever sequences have meanings different from in Windows — ctrl-s, ctrl-q, ctrl-d, ctrl-c, ctrl-z.

Slide 6

Shells, Continued

- Processing of what you type in:
  - Shell provides in-place editing arrow and other keys, command history, tab completion of filenames, etc. — until you press "return".
  - Shell then processes command line — expands wildcards and references to variables, "tokenizes" command into commandname and parameters.
  - Shell locates command in "search path" (`PATH` environment variable) and forks off a new process.
  - Command's return code then available via shell variable.

Slide 7

Processes and “Job Control”

- Normally, command you type is a “foreground process”. Append `&`, though, and you get a “background process”.
- Can make a foreground process a background process, and vice versa.
- Can even run commands in “batch” mode.

Slide 8

I/O Redirection

- In programming classes I talk about “reading from standard input” (`stdin`) rather than “reading from the keyboard”. Why?

  How about `stdout`, `stderr`?
- `stdin` can come from keyboard, file, or inline in shell script. `stdout` and `stderr` can go to terminal or file (overwrite or append), separately or together. (Syntax depends in part on which shell you’re using.)
- How is this useful? (e.g., in program development? testing?)
- *OR* — remember quotation from last time?

  “Write programs that do one thing and do it well. Write programs to work together. Write programs to handle text streams, because that is a universal interface.”

Slide 9

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: `sort`, `head`, `wc`, `sed`, `awk`, and too many others to name.

  Other programs that fit in well — `more`, `less`, `grep`.

Slide 10

Minute Essay

- None — sign in.