Unix Essentials

Devin J. Pohly <djpohly@cse.psu.edu>

The Unix Philosophy

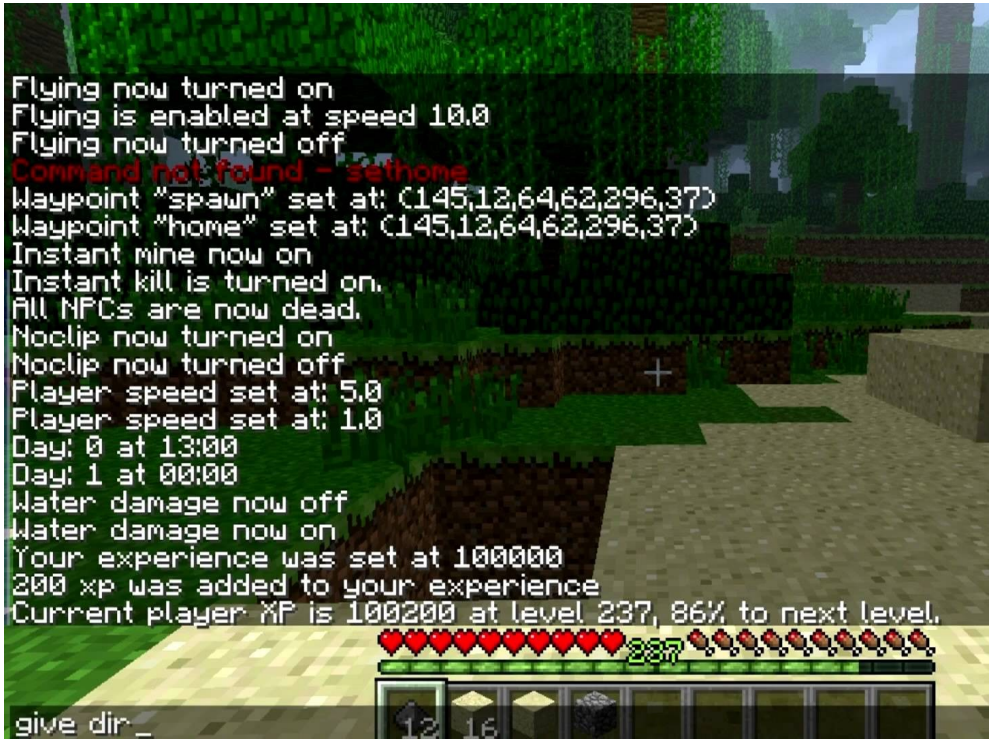
PENNSTATE 1855

- 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.

  - Doug McIlroy, Unix patriarch

Command line interface

- Command line? Why?
  - Efficient and powerful
  - Scriptable
  - Simple and reliable
    - Always works... even if everything else is b0rked!
- What is it?
  - Shell program ("bash" on Linux)
  - Interprets built-in commands
  - Runs other programs
  - Runs shell scripts



Getting started

- Boot the VM
- Log into the GUI
- Open Terminal Emulator
  - Under Accessories
  - Also on launch panel on the bottom of the screen
- You are now typing input for the shell

Common command syntax

Program:

```
ls
```

Program with an argument:

```
ls /home
```

Program with an option:

```
ls -l
```

Program with options and an argument:

```
ls -l -a /home
ls -l --all /home
ls -la /home
```

Standard filesystem layout

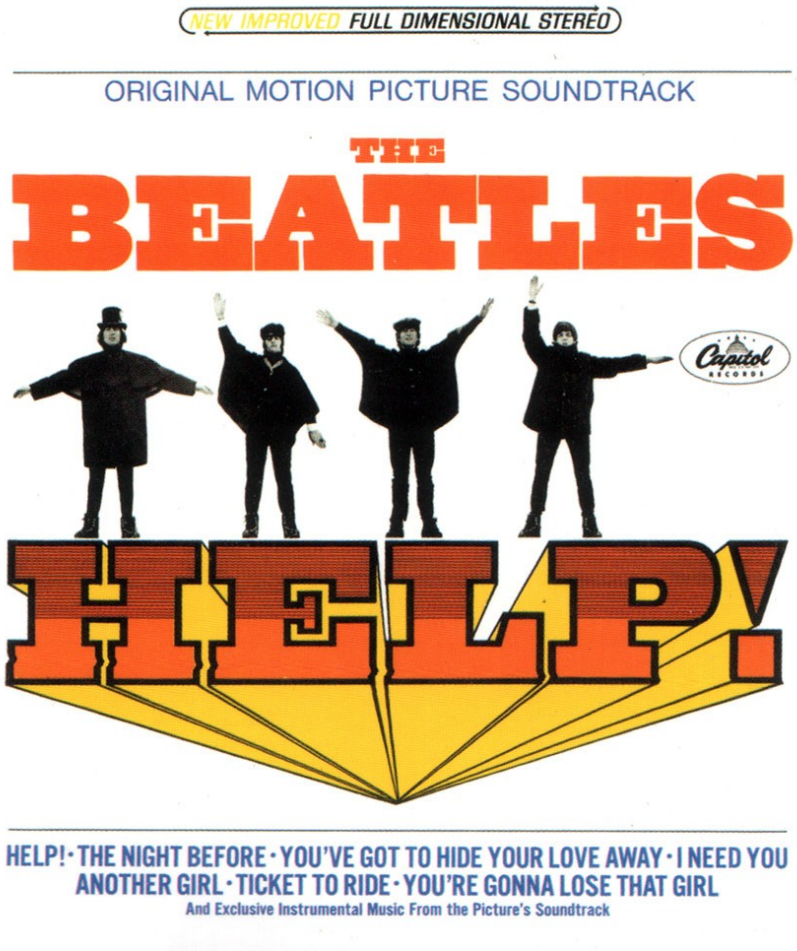
- Grouped by type
- `/usr`: installed software
  - `/usr/bin`, `/usr/lib`, ...
- `/etc`: configuration
- `/home`: users' own files
- `/dev`: devices
- `/tmp`: temporary files

Navigating the CLI

- Files
  - Listing: `ls`
  - Moving/copying: `mv`, `cp`
  - Deleting: `rm`
- Directories
  - "You are here": `pwd`
  - Moving around: `cd` *`DIR`*
  - Creating/removing: `mkdir`, `rmdir`
  - Special directory names
    - `.` means the current directory
    - `..` means the parent directory

Help!

- Built-in commands: `help`
- Everything else: `man`
  - Standard layout
  - `man operator`
  - `man ascii`
  - `man man`!
  - Get used to reading these!
- Keyword search: `man -k`
- (Also Google)



Editing files

- Unix philosophy again: text is important!
  - Configuration? Text files.
  - Scripting? Text files.
  - Programming? Text files.
- Vim (and `vi`)
  - Do one thing and do it well

Vim setup

```
sudo apt-get install vim
```

```
wget -U mozilla tiny.cc/311setup
```

```
sh 311setup
```

Understanding Vim

- "Modal" editor
  - Mode indicator in bottom-left corner
- Normal mode (default)
  - Navigate the file
  - Everything except typing text
  - When in doubt, Esc will put you in Normal mode.
- Insert mode
  - Typing text

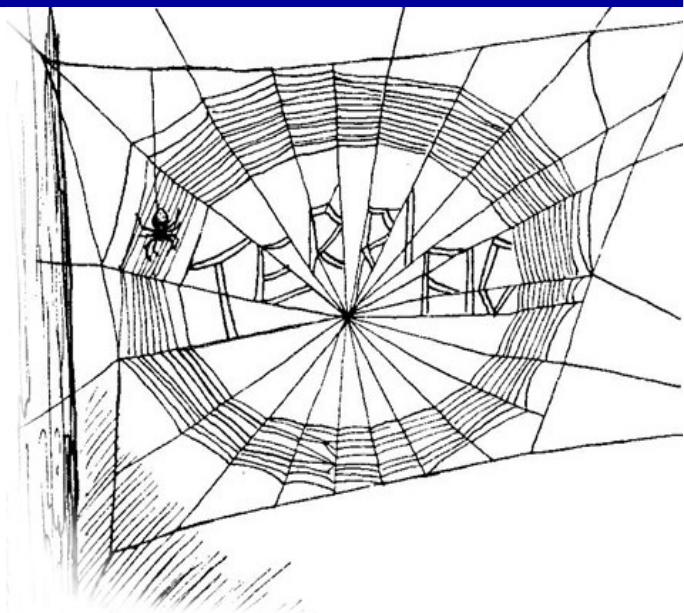
Vim: getting around

- Arrows optional!
  - Moving: h, j, k, l
  - Paging: Ctrl-f/b, Ctrl-u/d
  - Other keys work too
- Searching: /
  - (Regular expressions)
- Practice:
  `vim /etc/services`
- To quit, type `:q`, Enter in Normal mode.

Vim: making changes

- In Normal mode:
  - i to insert at cursor
  - A to append to line
  - o to open a line below
  - O to open a line above
  - u for unlimited undo!
- Esc: return to Normal
- Practice: `vim hello.c`
- Write and quit: `:wq`

Salutations!



- Make a file `hello.c`, containing:

```
#include <stdio.h>

int main(void)
{
        printf("yo dawg\n");
        printf("I mean hello world\n");
        return 0;
}
```

- Write and quit: `:wq`, Enter.
- It's OK if you don't understand the exact details.

Compile and run

- Run the compiler:

```
gcc -o hello hello.c
```

- Execute the program:

```
./hello
```

- Edit it again
  - Delete a printf by moving to it and pressing "dd".
  - Now add another printf by using the "o" command.
    - Notice that "o" puts you in Insert mode!
- Write, quit, compile, run!

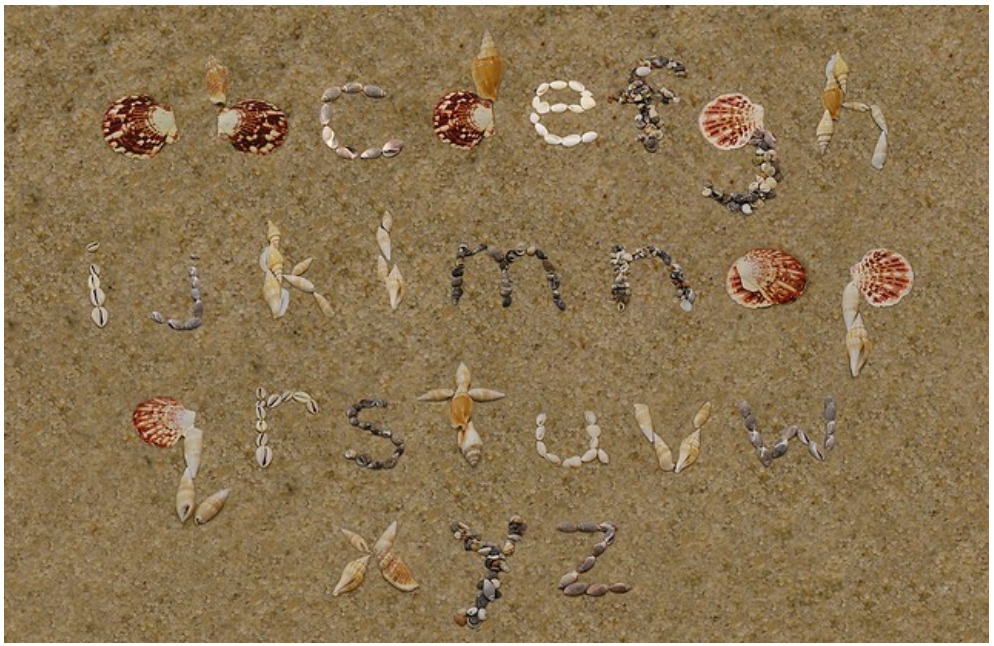
Shell scripting

- Getting tired of typing

  ```
  gcc -o hello hello.c
  ```

  yet?
- Shell script = list of commands to run
- Put this line in a file called `build.sh`.
- Now just run

  ```
  sh build.sh
  ```



Learning Vim

- Run `vimtutor` at the command line
  - Super simple introduction
  - I'd *highly* recommend going through 2 or 3 lessons
- `:help`
  - `:help user-manual`
- Many, many features
  - Find the things that work for you
- Practice!
  - GVim for Windows, MacVim for Mac