Shell Basics

Linux System Administration
Fall 2023

The command line

- You need to know your way around the CLI
  - When stuff breaks that may be all you have
  - Readily runs remotely via ssh
  - One liners can do complex things
- Scripting simplifies your life
  - Creates new custom commands
  - Laziness can be a virtue

System access basics for Lab 1

- root can do ***anything***
  - Root logins can be restricted
- User access by
  - UID - user id
  - GID - group id
    - users can belong to multiple groups
- File permissions control access to commands
- sudo - run as root
  - access by user or group
- Administering users
  - useradd
  - userdel
  - usermod
- Figure out how to allow privileged assess to your non-root login

The Unix ABCs

- A is for awk, which runs like a snail
- B is for biff, which reads all your mail
- C is for cc, as hackers recall
- D is for dd, the command that does all
- E is for emacs, which rebinds your keys
- F is for fsck, which rebuilds your trees
- G is for grep, a clever detective
- H is for halt, which may seem defective
- I is for indent, which rarely amuses
- J is for join, which nobody uses
- K is for kill, which makes you the boss
- L is for lex, which is missing from DOS
- M is for more, from which less was begot
- N is for nice, which really is not
- O is for od, which prints out things nice
- P is for passwd, which reads in strings twice
- Q is for quota, a Berkeley-type fable
- R is for ranlib, for sorting a table
- S is for spell, which attempts to belittle
- T is for true, which does very little
- U is for uniq, which is used after sort
- V is for vi, which is hard to abort
- W is for whoami, which tells you your name
- X is, well, X, of dubious fame
- Y is for yes, which makes an impression, and
- Z is for zcat, which handles compression.

Figuring stuff out

- man - online help
  - man sections
    - 1 User Commands
    - 2 System Calls
    - 3 C Library Functions
    - 4 Devices and Special Files
    - 5 File Formats and Conventions
- apropos (man -k)
  - and the SEE ALSO part of man pages
- -h flag on many commands

Shells

- Thompson Shell (1971)
- sh Bourne Shell (1977)
- csh C Shell (1978)
- ksh Korn Shell (1983)
- tcsh T Shell (1983)
- bash Bourne-again Shell (1989)

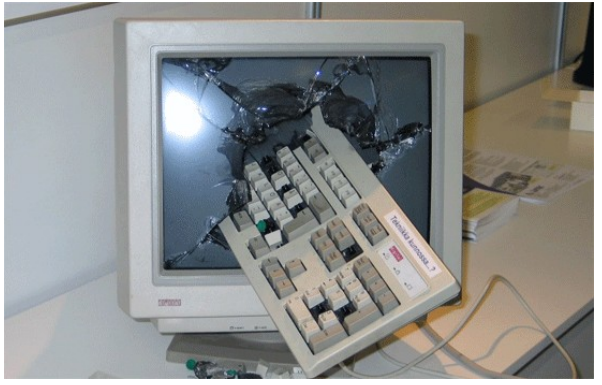
Anatomy of a shell

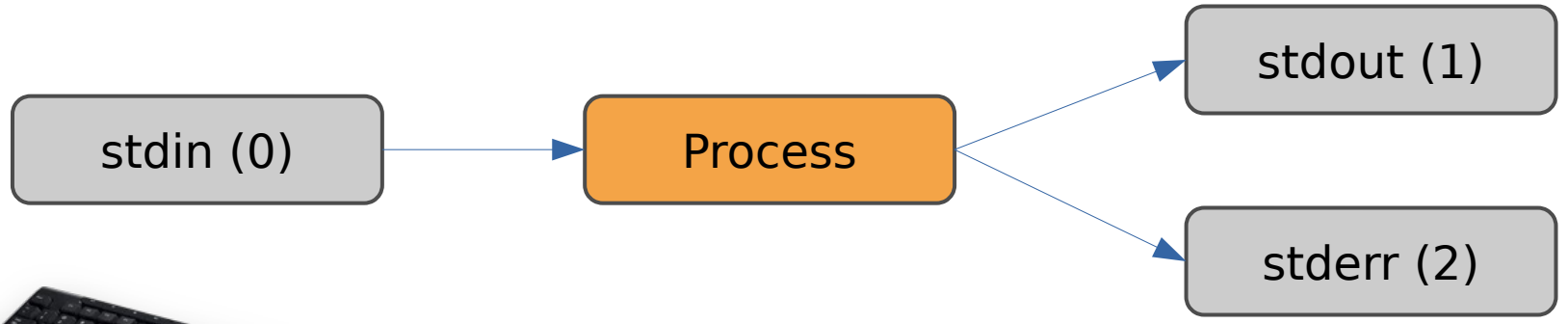
- user@host:dir# command arg1 arg2 arg3
  - Prompt (PS1)
  - Command expanded by shell
    - Wildcards and variables
- ; or \n separates commands
- Whitespace separates arguments
- Data can be redirected with <, > and |

Unix process (commands)









Redirecting stdout

>	Write output to file
>>	Append output to file
|	Pipe output to next program

Redirecting stdin

< Read input from file

```
mail vlakkies@colorado.edu < message.txt
```

<< Read input until tag

```
mail vlakkies@colorado.edu << EOF
This is the text
of this email.
EOF
```

<<< Read input from current line

```
mail vlakkies@colorado.edu <<< 'This is the text'
mail vlakkies@colorado.edu <<< "Mail from $PWD"
```

Special Redirection

2>	Redirect stderr
2>&1	Merge stderr into stdout

Converting stdio to arguments

- `xargs`
  - Change stdin to command line arguments
- `grep -l -dskip foo *|xargs sed -i s/foo/bar/`
  - `grep`
    - -l only list file named
    - -dskip ignore directories
  - `sed`
    - change `foo` to `bar` in place (`-i`)

curses (ncurses)

- Text-based API used by CLI programs
  - top, screen, vim, cping, Midnight Commander, ...
- ncurses (and other) re-implementation
- Works very well over ssh connections
- Uses terminfo to get terminal capabilities

Variables

- Set syntax depends on shell
  - csh: `set x=1`
  - bash: `x=1`
- Usage
  - `echo "x=$x"`
- Promoting to global
  - `export x`

Quotes

- Single quote (') - no variable expansion
  - `echo 'This string is dollar x: $x'`
- Double quote (") - expand variables
  - `echo "dollar x equals $x"`
- Back quote (`) - execute as command
  - `vi `grep –l 'Das Foo' *.txt``

Editors

- sed - streaming editor
- vi - visual editor
  - vim - Vi IMproved
- nano
- emacs

Stuff to know

- PATH Where to find commands
  - `which` *`command`*
- Special directories: `.` `..` `~`
- Dot files and directories
- Example filenames: foo bar
- Sourcing files: `. script`

Commonly used commands

- ls - list
- cp - copy
- mv - move
- rm - remove
- directory commands
  - cd - change
  - mkdir - make
  - rmdir - remove
- echo - write to screen
- cat - concatenate
- more - paginate
- grep - search
- xargs - convert to arguments
- ssh - remote shell
- scp - remote copy
- rsync - remote sync

Process management

- `Ctrl-C` terminate the current task
- `Ctrl-Z` suspend the current task
- `bg` resume suspended task in the background
- `fg` resume task in foreground

Commands for working with text

- `wc` - word/line counts
- `sort` - ordering
- `uniq` - repeated lines
- `cut` - show columns
- `paste` - merge files
- `join` - merge on field

- Streaming editors
  - `sed`
  - `awk`
  - `tr`
- Comparing files
  - `diff`
  - `sdiff`
  - `cmp`

System Commands 1

- Processes
  - `who`
  - `ps`
  - `top`
  - `kill`
  - `renice`
- Networking
  - `ip addr`
    `ip route`
  - `ping`
  - `traceroute`
  - `dig`
  - `nslookup`

System commands 2

- File system
  - `fsck`
  - `mount`
  - `umount`
  - `fdisk`
  - `mkfs`
  - `df`
  - `du`
  - `sync`
- File manipulation
  - `chmod`
  - `chown`
  - `touch`
  - `truncate`
- Compression
  - `gzip, gunzip`
    `bzip2, bunzip2`
  - `zcat, zgrep, zmore, zdiff`
- Swiss army chainsaw copier
  - `dd`

Document management

- LibreOffice (WYSIWYG)
  - `lowriter, localc, loimpress,lodraw,...`
- Typesetting
  - `troff, nroff, groff, ...`
  - `TeX, LaTeX, pdfTex, pdfLaTex, ...`