

Lecture 02: user environment it's a series of tubes

Hands-On Unix System Administration DeCal

2012-09-10

Review

- ❖ Shell interaction
- ❖ Basic commands

Shell variables

Shell expansion

Pipes

Review

Shell interaction

Review

❖ Shell
interaction

❖ Basic
commands

Shell variables

Shell expansion

Pipes

- e.g., `bash`
- there's a prompt, type something, read output, repeat
- command history
- tab-completion

Basic commands

Review

❖ Shell
interaction

❖ Basic
commands

Shell variables

Shell expansion

Pipes

- `pwd`, `ls`, `cd`, `mkdir`, `rmdir`,
`touch`, `mv`, `cp`, `rm`, `echo`, `cat`,
`less`, `grep`
- `whoami` (**who** am I)
`uname` (**what** I am I running)
`hostname` (**where** am I)
`date` (**when**)

Review

Shell variables

- ❖ Variables
- ❖ Shell script arguments
- ❖ Environment variables
- ❖ Env examples
- ❖ Quotes

Shell expansion

Pipes

Shell variables

Variables

Review

Shell variables

❖ Variables

❖ Shell script arguments

❖ Environment variables

❖ Env examples

❖ Quotes

Shell expansion

Pipes

- case-sensitive
- quotes matter (more on this later)
- `$ variable='value'`
`$ echo "$variable"`
`value`

Shell script arguments

Review

Shell variables

❖ Variables

❖ Shell script arguments

❖ Environment variables

❖ Env

examples

❖ Quotes

Shell expansion

Pipes

- **\$1, \$2, \$3...**: individual arguments
- **\$@, \$***: all arguments
(behave differently quoted)
- **\$#**: number of arguments
- **\$0**: name of running script
- **\$?**: exit status of previous command

Environment variables

Review

Shell variables

- ❖ Variables
- ❖ Shell script arguments

❖ Environment variables

- ❖ Env examples
- ❖ Quotes

Shell expansion

Pipes

- created by init scripts (during startup), by shell, or other parent processes along the way
 - ◆ inherited by child processes
 - ◆ typically names are capitalized
- display with `env` or `printenv`, set with `export` (for `bash` only)

Env examples

Review

Shell variables

- ❖ Variables
- ❖ Shell script arguments
- ❖ Environment variables

❖ **Env examples**

❖ Quotes

Shell expansion

Pipes

- **\$HOME** (home directory): e.g.,
`/home/jdoe`
- **\$PATH** (directories to search for commands): e.g.,
`/usr/local/bin:/usr/bin:/bin`
- **\$PS1** (prompt display, depends on shell): e.g., `\n\[\e[0;31m\]\u\[\e[m\]`
`@\[\e[1;34m\]\w \[\e[2;90m\]\@`
`\n\[\e[m\]\[\e[0;35m\]\h\[\e[m\]\[\e[0;31m\]\$`
`\[\e[m\]\[\e[0;32m\]`

Quotes

Review

Shell variables

- ❖ Variables
- ❖ Shell script arguments
- ❖ Environment variables
- ❖ Env examples
- ❖ **Quotes**

Shell expansion

Pipes

- **no quotes:** spaces and “special” characters (`$`, `*`, `?`, `!`, `\`, `\`, `'`, `"`, etc.) have special meanings
- **double quotes:** double quote and (`$`, `\`, `\`) have special meanings
- **single quotes:** only single quote (`'`) has special meaning

Review

Shell variables

Shell expansion

- ❖ Aliases
- ❖ Globbing
- ❖ Substitution

Pipes

Shell expansion

Aliases

Review

Shell variables

Shell expansion

❖ Aliases

❖ Globbing

❖ Substitution

Pipes

- replacement of word by another string
- e.g., in `bash`, `alias dir='ls'`
- view current aliases with `alias`
- **alternatives:** functions or scripts/symlinks in `PATH`

Globbering

Review

Shell variables

Shell expansion

❖ Aliases

❖ **Globbering**

❖ Substitution

Pipes

- parameter expansion using wildcard character (* and ?)
- prevent with escape characters or quotes

Substitution

Review

Shell variables

Shell expansion

❖ Aliases

❖ Globbing

❖ **Substitution**

Pipes

- output (stdout) of command in backticks (`) is substituted in command line
- useful to assign variables, e.g.,
`USER= `whoami ``
- bash alternative `$ (. . .)`

Review

Shell variables

Shell expansion

Pipes

- ❖ Real pipes
- ❖ Silly words
- ❖ Wise words
- ❖ Unix pipes
- ❖ Standard streams
- ❖ Named pipes (FIFO)
- ❖ Network pipes (netcat)

Pipes



President Obama at TransCanada Stillwater Pipe Yard
(March 22 photo by AP via [bloomberg.com](https://www.bloomberg.com))

Silly words

Review

Shell variables

Shell expansion

Pipes

❖ Real pipes

❖ **Silly words**

❖ Wise words

❖ Unix pipes

❖ Standard streams

❖ Named pipes (FIFO)

❖ Network pipes (netcat)

“And again, the Internet is not something that you just dump something on. It’s not a big truck. It’s a **series of tubes.**”

(Sen Ted Stevens, R-AK)

Wise words

Review

Shell variables

Shell expansion

Pipes

- ❖ Real pipes
- ❖ Silly words
- ❖ **Wise words**
- ❖ Unix pipes
- ❖ Standard streams
- ❖ Named pipes (FIFO)
- ❖ Network pipes (netcat)

“This is the Unix philosophy. 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, inventor of Unix pipes)

Unix pipes

Review

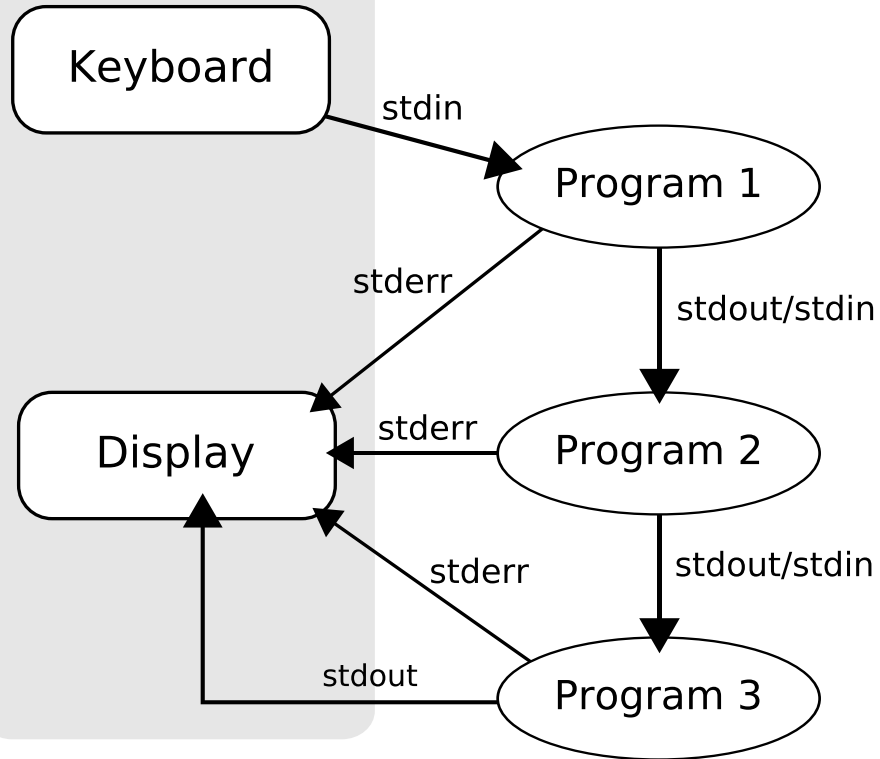
Shell variables

Shell expansion

Pipes

- ❖ Real pipes
- ❖ Silly words
- ❖ Wise words
- ❖ **Unix pipes**
- ❖ Standard streams
- ❖ Named pipes (FIFO)
- ❖ Network pipes (netcat)

Text terminal



<http://en.wikipedia.org/wiki/File:Pipeline.svg>

Standard streams

Review

Shell variables

Shell expansion

Pipes

- ❖ Real pipes
- ❖ Silly words
- ❖ Wise words
- ❖ Unix pipes
- ❖ **Standard streams**
- ❖ Named pipes (FIFO)
- ❖ Network pipes (netcat)

- standard input (**stdin**): text going into program (read), typically TTY unless redirected
- standard output (**stdout**): text going out of program (write), typically TTY unless redirected
- standard error (**stderr**): error/diagnostic text output, not redirected

Named pipes (FIFO)

Review

Shell variables

Shell expansion

Pipes

- ❖ Real pipes
- ❖ Silly words
- ❖ Wise words
- ❖ Unix pipes
- ❖ Standard streams
- ❖ **Named pipes (FIFO)**
- ❖ Network pipes (netcat)

- named in contrast to anonymous ones created by shell
- special file created using `mkfifo`
- can be used as a temporary file

Network pipes (netcat)

Review

Shell variables

Shell expansion

Pipes

- ❖ Real pipes
- ❖ Silly words
- ❖ Wise words
- ❖ Unix pipes
- ❖ Standard streams
- ❖ Named pipes (FIFO)
- ❖ Network pipes (netcat)

Imagine the possibilities:

- dd
- gzip