

# Ling 555 — Programming for Linguists

Robert Albert Felty

Speech Research Laboratory  
Indiana University

Sep. 03, 2008

# How programming will make your life easier

An easier life

Unix basics

help / options

Philosophy

Resources

## Example

I recently discovered that a portion of the sound files from the TIMIT database were grossly distorted. There are 6300 files. How can I remove all the distorted ones?

```
# this snippet removes all clipped files, by checking
the output from sox
for file in `find . -name "*.wav" -print`; do
  if [[ `sox $file -n stat 2>&1 | grep -E
    "^(Try:|Can't|(Min|Max)imum
    amplitude:\s+-?1\.\00)"` ]]; then echo "$file
    CLIPPED";
    mv $file $file.clipped;
  fi;
done
```

# Level Check

**An easier life**

**Unix basics**

navigation

The PATH

permissions

Reading files

Pipes and streams

**help / options**

**Philosophy**

**Resources**

How many people:

- 1 Have ever used the terminal (command line)?
- 2 Know some basic commands like `cd` `ls` `cp` `mv`?
- 3 Fully understood the previous example?

# Filesystem navigation

An easier life

Unix basics

navigation

The PATH

permissions

Reading files

Pipes and streams

help / options

Philosophy

Resources

**cd** Change directory to *foo*. Without arguments, changes to your home directory

**ls** list the contents of directory *foo*. Without arguments, list the contents of the current directory

**pwd** print out the current working directory (cwd)

**mv** Move a file to a different location

**cp** Make a copy of a file in a different location

**rm** Remove a file (deletes permanently)

**mkdir** Create a directory

**touch** Create an empty file

# Handy shortcuts

An easier life

Unix basics

navigation

The PATH

permissions

Reading files

Pipes and streams

help / options

Philosophy

Resources

## Home directory shortcut

`~` is a shortcut for your home directory. E.g., your Desktop is located at `~/Desktop`

## TAB completion

If you start typing a command or filename, then press TAB, the shell will complete the word for you

## Command history

The shell keeps a history of your commands. To scroll through them, simply press the up key. For more info on command history, including how to search it, see: <http://www.catonmat.net/blog/the-definitive-guide-to-bash-command-line-history/>

# Finding the right PATH

An easier life

Unix basics

navigation

The PATH

permissions

Reading files

Pipes and streams

help / options

Philosophy

Resources

## Definition

Whenever you type a command in a unix-like shell, the shell searches through the path to see if it can find such a command. You will want to make sure commonly used programs are used in your path. For example, `ls` is normally located in `/bin/ls`, but since `/bin` is in your path, you don't have to type the whole path (but you can of course).

## PATH search order

If you have two programs named *foo*, one in `/usr/local/bin`, and one in `/bin`, whichever one comes first in your path will be used when you simply type `foo`. If you want to make sure that a particular one is used, specify the entire path.

# Showing and changing your path

An easier life

Unix basics

navigation

The PATH

permissions

Reading files

Pipes and streams

help / options

Philosophy

Resources

To show your path:

```
echo $PATH
```

To change your path:

```
export PATH="/some/new/path:${PATH}"
```

Search the path for a program:

```
which foo
```

# File permissions

An easier life

Unix basics

navigation

The PATH

permissions

Reading files

Pipes and streams

help / options

Philosophy

Resources

## Definition

Every file on a UNIX system has 3 sets of permissions, specifying who can read, write, and execute the file. The three sets apply to the user, group, and others

## Example

```
drwxr-xr-x  4 robfelty root      4096 Jul 10 23:02
             fender4star
-rwxr-xr-x  1 robfelty robfelty 1137 Aug 19 14:12
             syncWithDreamhost
lrwxrwxrwx  1 robfelty yootlers    21 Jun  9 10:43
             images -> ../fediblety/images/
```



# File permissions

An easier life

Unix basics

navigation

The PATH

permissions

Reading files

Pipes and streams

help / options

Philosophy

Resources

## Example

To change file permissions, you can use the commands  
chown, chgrp, chmod

Change the user to john for the file johnsfile.txt

```
chown john johnsfile.txt
```

Change the group to johndandmary for the file  
johnsfile.txt

```
chgrp johndandmary johnsfile.txt
```

Make a file executable by all users

```
chmod a+x myFirstPythonScript.py
```

# Reading files

An easier life

Unix basics

navigation

The PATH

permissions

Reading files

Pipes and streams

help / options

Philosophy

Resources

**cat** Prints out an entire file (or files)

**tac** Prints out an entire file backwards (line by line)

**head** Prints out first  $n$  lines of a file (default 10)

**tail** Prints out last  $n$  lines of a file (default 10)

**wc** Counts number of lines in a file

**nl** Prints out line numbers for a file

**cut** Prints out particular columns of a file

**paste** Pastes together files column-wise

**split** Splits a file into multiple files, each with  $n$  lines (default 1000)

**more** Interactively print out file, one screen worth at a time

**less** Fancier version of more. Less is more.

# Sorting files et al.

**sort** Sort a file (many options)

**uniq** Print unique lines from a sorted input

**diff** Compare the contents of 2 files

## Example

Sort the CELEX file by frequency (primary key) with highest frequency coming first, and orthography (secondary key), ignoring case, and save the results in a new file

```
sort -t '\ ' -k 3,3rn -k 2,2fd celex.cd >  
celex.sorted
```

An easier life

Unix basics

navigation

The PATH

permissions

Reading files

Pipes and streams

help / options

Philosophy

Resources

# Put this in your pipe and ...

An easier life

Unix basics

navigation

The PATH

permissions

Reading files

Pipes and streams

help / options

Philosophy

Resources

## Definition

You can pipe the output from one program into another program

## Example

To display only the first 10 entries of a directory listing, you could do:

```
ls | head
```

How would you display the last 10?

## Example

Create a histogram of frequencies from the CELEX

```
cut -f 3 -d '\ ' celex.cd | sort -rn | uniq -c
```

# Three streams

An easier life

Unix basics

navigation

The PATH

permissions

Reading files

Pipes and streams

help / options

Philosophy

Resources

## Standard input (STDIN)

The input to a program. Can be specified like so:

```
cat < foo
```

## Standard output (STDOUT)

The output from a program. Can be redirected to a file like so:

```
ls > foo.txt
```

To append to a file, use:

```
ls >> foo.txt
```

## Standard error (STDERR)

Error messages are sent to a different output stream.

You can redirect stderr like so:

```
ls 2> error.txt
```

# Subshells

An easier life

Unix basics

navigation

The PATH

permissions

Reading files

Pipes and streams

help / options

Philosophy

Resources

## Definition

You can run a command in a subshell by using backticks ``foo``. The result of the command can be stored and used.

## Example

Remove file extensions and preceding path elements from a filename

```
echo `basename l55practiceFiles/a.txt .txt`
```

# Options

Most UNIX commands have a variety of options which can be specified on the command line. Most programs allow two different types of options:

**short options** e.g. `-h` = “print help for this command”.  
You can group these together, like `-hv`, means, print verbose help

**long options** e.g. `--help`. Long options take longer to type, but obviously are more descriptive

**other** Some commands take long-style options with a single dash, notably `java`.

## Example

Some options also take arguments. To print the first 25 lines of a file, we could do:

```
head -n 25 foo
```

An easier life

Unix basics

help / options

Practice

Philosophy

Resources

# Help

An easier life

Unix basics

help / options

Practice

Philosophy

Resources

Most standard UNIX commands are quite well documented. To get help on a particular command, try:

```
man foo
```

Moving around a manual page: (same commands as less by default)

`<space>` Go forward one screen

`↑ ↓` Navigate with arrow keys

`k j` Navigate from the home row

`/` Search for a word (can use regular expressions)

`n` Go to next instance of found word

`N` Go to previous instance of found word

`q` Exit the manual pager



# Practice

An easier life

Unix basics

help / options

Practice

Philosophy

Resources

Download practiceFiles.zip from oncourse and unzip it into your home directory.

- 1 Display lines (files) 11–20 from a directory listing of 1555practiceFiles

```
ls | head -n 20 | tail OR
```

```
ls | tail -n +11 | head
```

- 2 Produce a numbered list of the files in 1555practiceFiles

```
ls | nl
```

- 3 Count the total number of characters in the filenames in 1555practiceFiles

```
ls | wc -c
```

- 4 Move files 11-20 to a new directory tmp

```
mkdir tmp
```

```
mv `ls | head -n 20 | tail` tmp
```

# Keep it simple, stupid!

**An easier life**

**Unix basics**

**help / options**

**Philosophy**

KISS

TMTOWTDI

The three virtues

**Resources**

## Definition

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

# TMTOWTDI

An easier life

Unix basics

help / options

Philosophy

KISS

TMTOWTDI

The three virtues

Resources

## Theorem

*The Perl motto is “There’s more than one way to do it.”  
Divining how many more is left as an exercise to the reader.*

# The three virtues

An easier life

Unix basics

help / options

Philosophy

KISS

TMTOWTDI

The three virtues

Resources

The three principal virtues of a programmer are:

- ① Laziness
- ② Impatience
- ③ Hubris

— Larry Wall, from *Programming Perl, 2nd. ed.* (also known as the Camel book)

# The three virtues

The three principal virtues of a programmer are:

① Laziness

The quality that makes you go to great effort to reduce overall energy expenditure. It makes you write labor-saving programs that other people will find useful, and document what you wrote so you don't have to answer so many questions about it. Hence, the first great virtue of a programmer. Also hence, this book. See also impatience and hubris. (p.609)

② Impatience

③ Hubris

— Larry Wall, from *Programming Perl, 2nd. ed.* (also known as the Camel book)

An easier life

Unix basics

help / options

Philosophy

KISS

TMTOWTDI

The three virtues

Resources

# The three virtues

The three principal virtues of a programmer are:

① Laziness

② Impatience

The anger you feel when the computer is being lazy. This makes you write programs that don't just react to your needs, but actually anticipate them. Or at least pretend to. Hence, the second great virtue of a programmer. See also laziness and hubris. (p.608)

③ Hubris

— Larry Wall, from *Programming Perl, 2nd. ed.* (also known as the Camel book)

An easier life

Unix basics

help / options

Philosophy

KISS

TMTOWTDI

The three virtues

Resources

# The three virtues

An easier life

Unix basics

help / options

Philosophy

KISS

TMTOWTDI

The three virtues

Resources

The three principal virtues of a programmer are:

- 1 Laziness
- 2 Impatience
- 3 Hubris

Excessive pride, the sort of thing Zeus zaps you for. Also the quality that makes you write (and maintain) programs that other people won't want to say bad things about. Hence, the third great virtue of a programmer. See also laziness and impatience. (p.607)

—Larry Wall, from *Programming Perl, 2nd. ed.* (also known as the Camel book)

# Additional Resources

An easier life

Unix basics

help / options

Philosophy

Resources

**unix ref** [http://www.cumc.columbia.edu/computers/html/unix/unix1\\_01.htm](http://www.cumc.columbia.edu/computers/html/unix/unix1_01.htm)

**unix ref** [http://infohost.nmt.edu/tcc/help/unix/unix\\_-cmd.html](http://infohost.nmt.edu/tcc/help/unix/unix_-cmd.html)

**delicious** <http://delicious.com/robfelty/1555>