

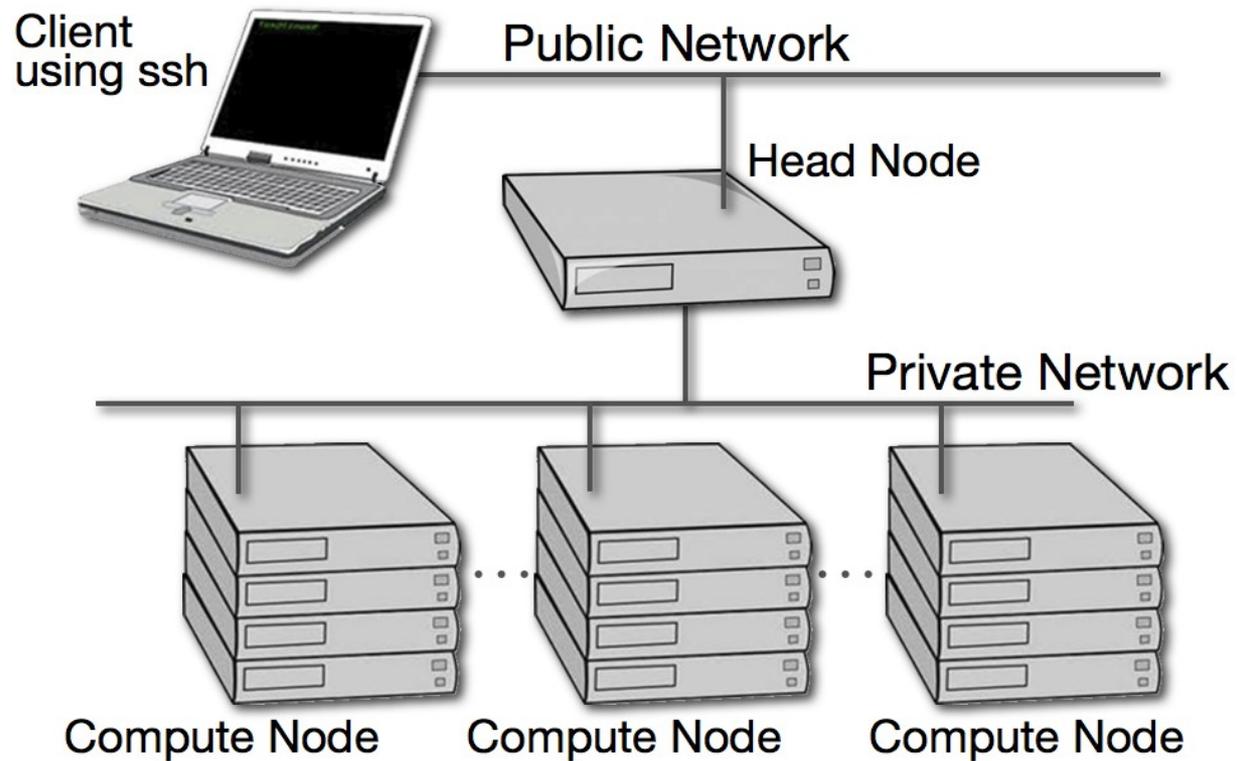
Part I

# **UNIX Workshop Series: Quick-Start**

# Objectives

- Overview – Connecting with ssh
- Command Window Anatomy
- Command Structure
- Command Examples
- Getting Help
- Files and Directories
- Wildcards, Redirection and Pipe
- Create and edit files

# Overview



# Connecting with ssh

- Open a Terminal program

- Mac: ***Applications > Utilities > Terminal***

- ```
ssh -Y username@centos.css.ude1.edu
```

- Linux: In local shell

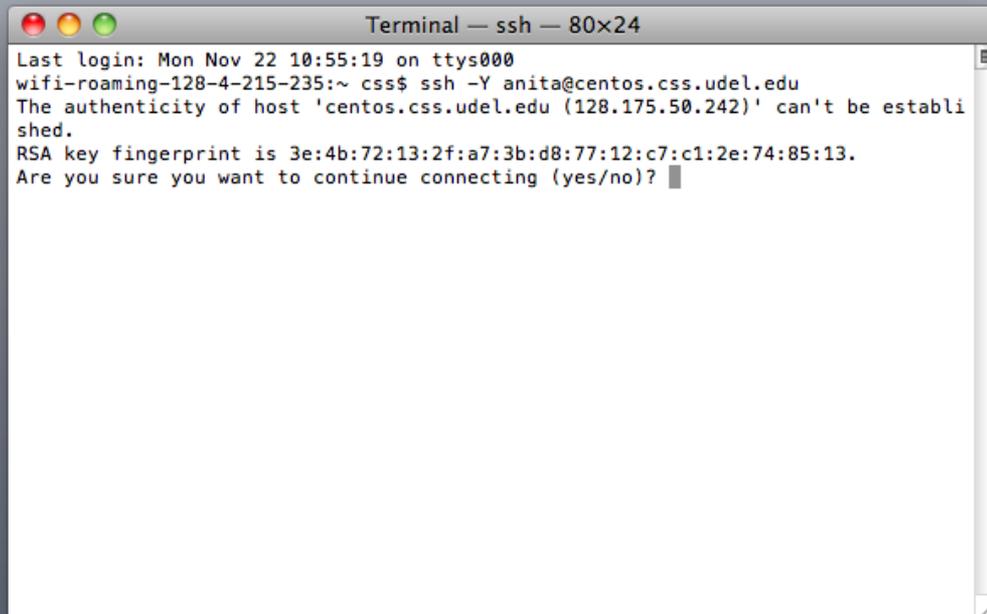
- ```
ssh -Y username@centos.css.ude1.edu
```

- Windows: Start Xming and PuTTY

- Create a saved session for the remote host name `centos.css.ude1.edu` using **username**

# Connecting with ssh

## □ First time you connect



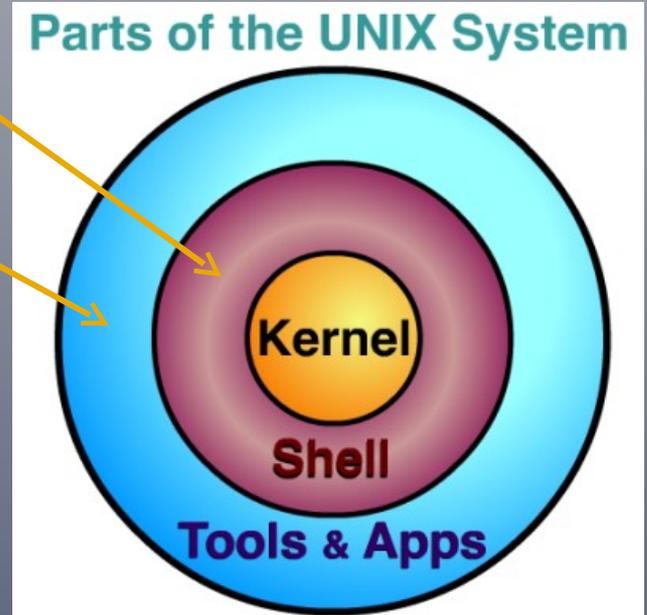
Terminal — ssh — 80x24

```
Last login: Mon Nov 22 10:55:19 on ttys000
wifi-roaming-128-4-215-235:~ css$ ssh -Y anita@centos.css.udel.edu
The authenticity of host 'centos.css.udel.edu (128.175.50.242)' can't be established.
RSA key fingerprint is 3e:4b:72:13:2f:a7:3b:d8:77:12:c7:c1:2e:74:85:13.
Are you sure you want to continue connecting (yes/no)? █
```

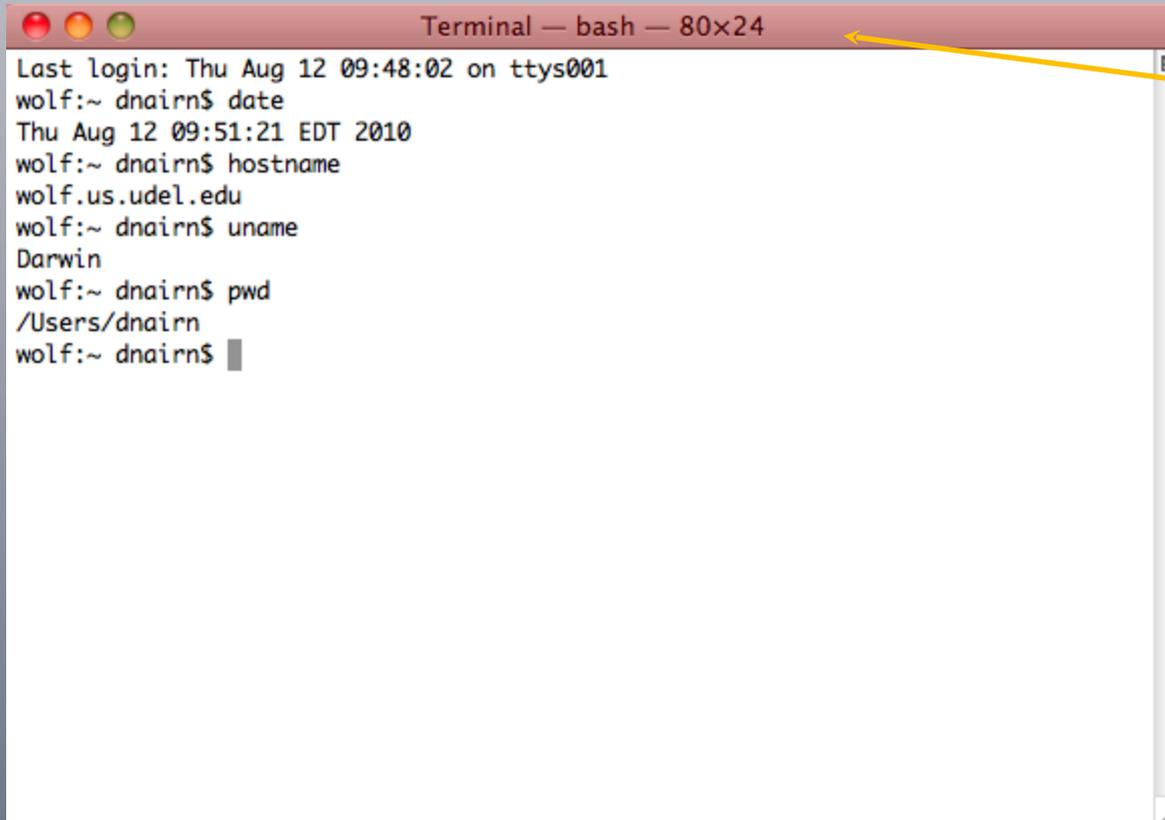


# Unix Basics

- Multi-user
- Case-sensitive
- Bash shell, command-line
- Commands



# Command Window Anatomy

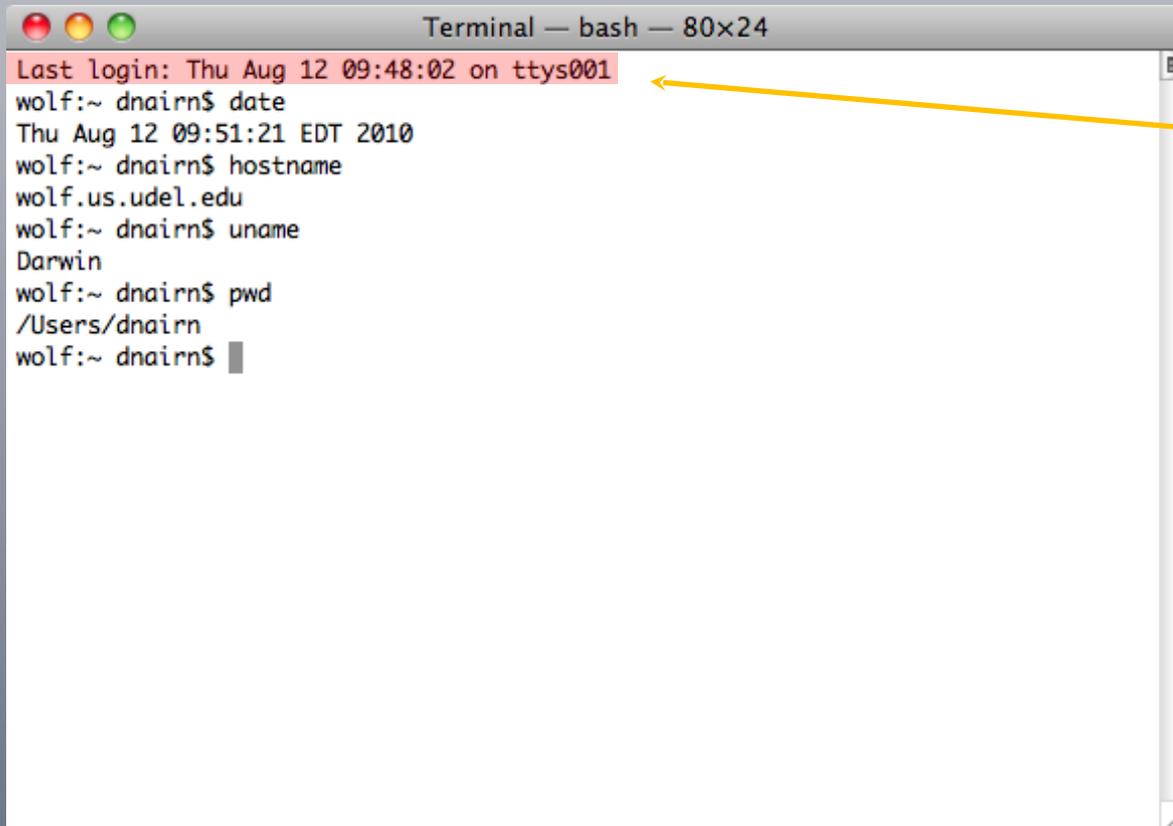
A screenshot of a macOS Terminal window. The title bar at the top is red and contains the text "Terminal — bash — 80x24". Below the title bar, the terminal displays the following text: "Last login: Thu Aug 12 09:48:02 on ttys001", "wolf:~ dnairn\$ date", "Thu Aug 12 09:51:21 EDT 2010", "wolf:~ dnairn\$ hostname", "wolf.us.udel.edu", "wolf:~ dnairn\$ uname", "Darwin", "wolf:~ dnairn\$ pwd", "/Users/dnairn", and "wolf:~ dnairn\$". A yellow arrow points from the "Title bar" label to the title bar of the terminal window.

```
Terminal — bash — 80x24
Last login: Thu Aug 12 09:48:02 on ttys001
wolf:~ dnairn$ date
Thu Aug 12 09:51:21 EDT 2010
wolf:~ dnairn$ hostname
wolf.us.udel.edu
wolf:~ dnairn$ uname
Darwin
wolf:~ dnairn$ pwd
/Users/dnairn
wolf:~ dnairn$
```

Title  
bar

Click in the title bar to bring the window to the front and make it active.

# Command Window Anatomy

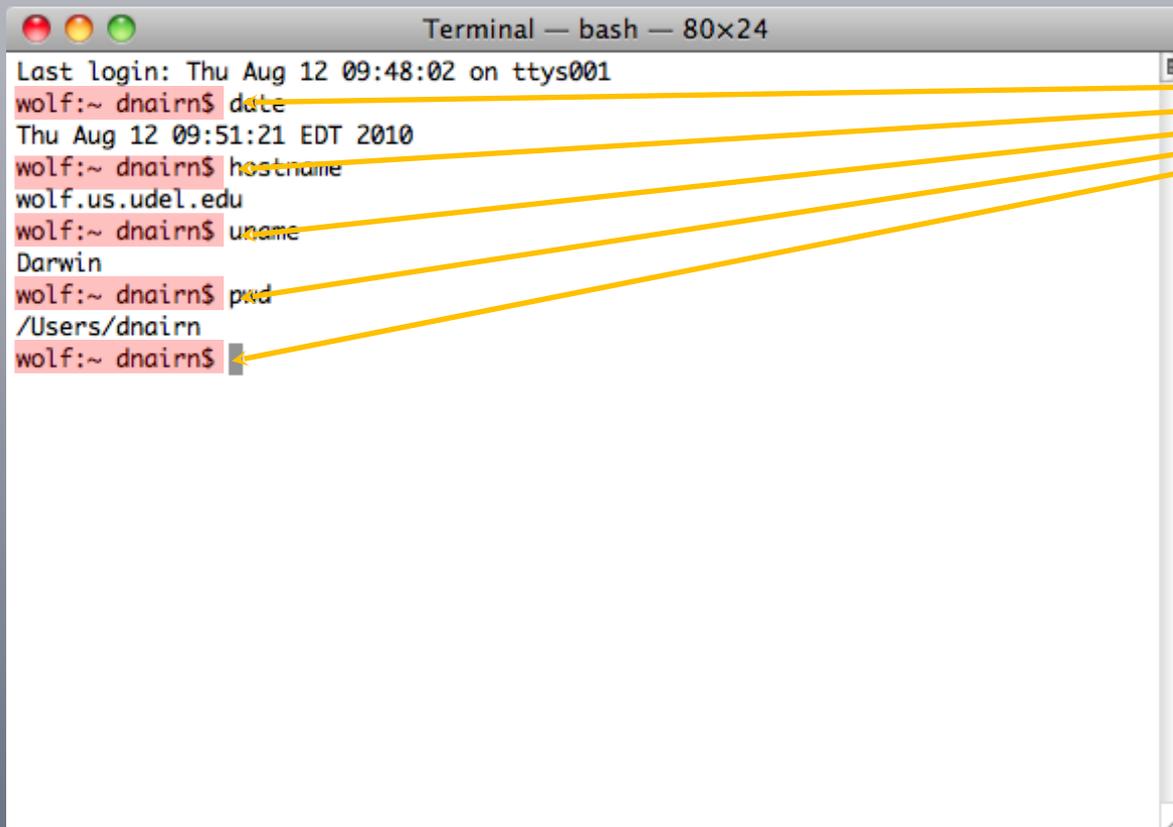


```
Terminal — bash — 80x24
Last login: Thu Aug 12 09:48:02 on ttys001
wolf:~ dnairn$ date
Thu Aug 12 09:51:21 EDT 2010
wolf:~ dnairn$ hostname
wolf.us.udel.edu
wolf:~ dnairn$ uname
Darwin
wolf:~ dnairn$ pwd
/Users/dnairn
wolf:~ dnairn$
```

Login banner

Appears as the first line of a login shell.

# Command Window Anatomy

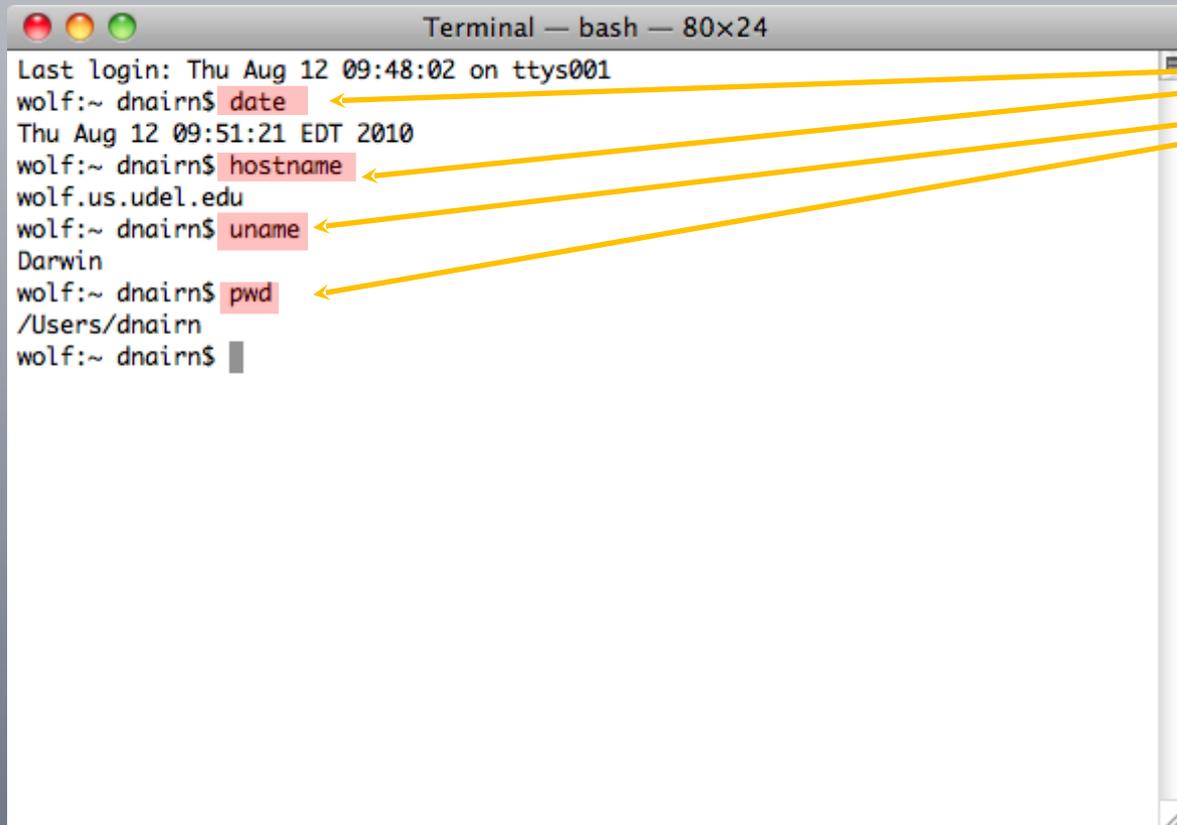


```
Terminal — bash — 80x24
Last login: Thu Aug 12 09:48:02 on ttys001
wolf:~ dnairn$ date
Thu Aug 12 09:51:21 EDT 2010
wolf:~ dnairn$ hostname
wolf.us.udel.edu
wolf:~ dnairn$ uname
Darwin
wolf:~ dnairn$ pwd
/Users/dnairn
wolf:~ dnairn$
```

Prompts

Appears at the beginning of a line and usually ends in \$.

# Command Window Anatomy



```
Terminal — bash — 80x24
Last login: Thu Aug 12 09:48:02 on ttys001
wolf:~ dnairn$ date
Thu Aug 12 09:51:21 EDT 2010
wolf:~ dnairn$ hostname
wolf.us.udel.edu
wolf:~ dnairn$ uname
Darwin
wolf:~ dnairn$ pwd
/Users/dnairn
wolf:~ dnairn$
```

Command  
input

Place to type  
commands, which  
may have options  
and/or arguments.

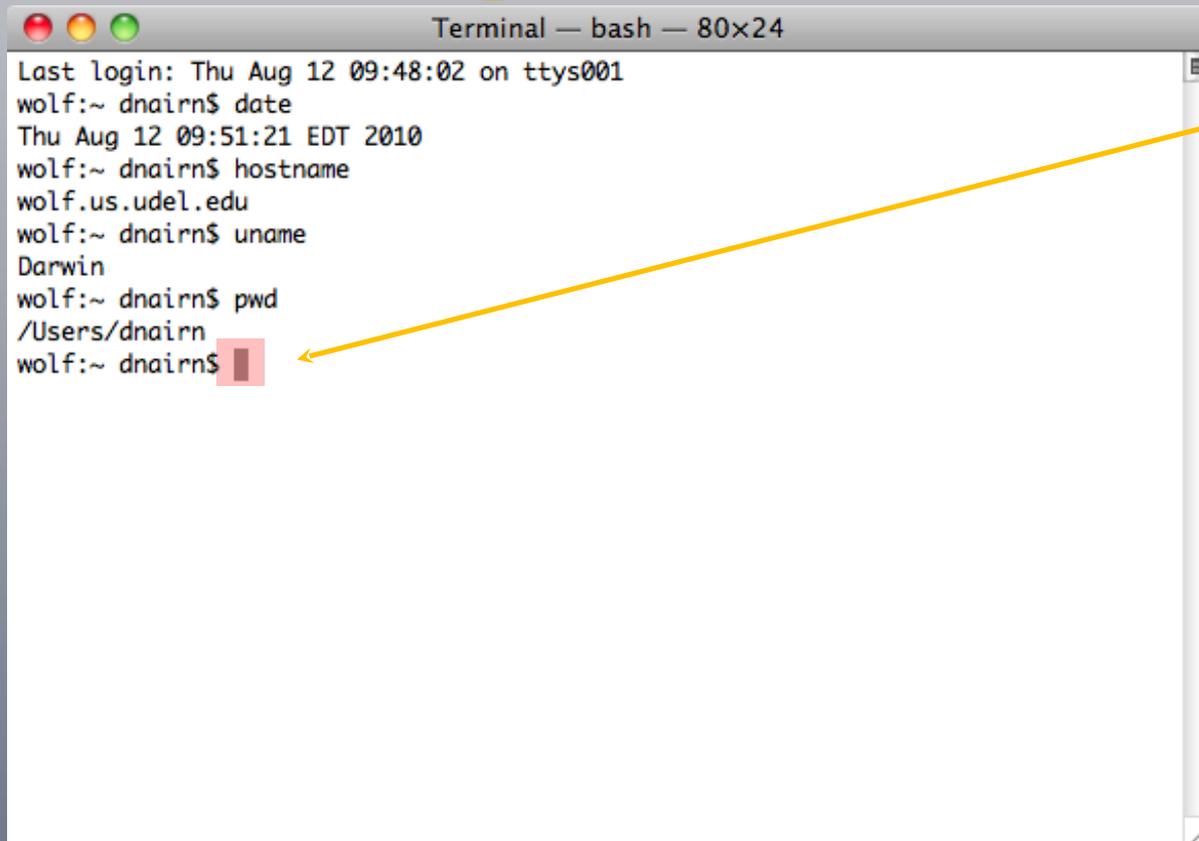
# Command Window Anatomy

```
Terminal — bash — 80x24
Last login: Thu Aug 12 09:48:02 on ttys001
wolf:~ dnairn$ date
Thu Aug 12 09:51:21 EDT 2010
wolf:~ dnairn$ hostname
wolf.us.udel.edu
wolf:~ dnairn$ uname
Darwin
wolf:~ dnairn$ pwd
/Users/dnairn
wolf:~ dnairn$
```

Command output

Place for command response, which may be many lines long.

# Command Window Anatomy

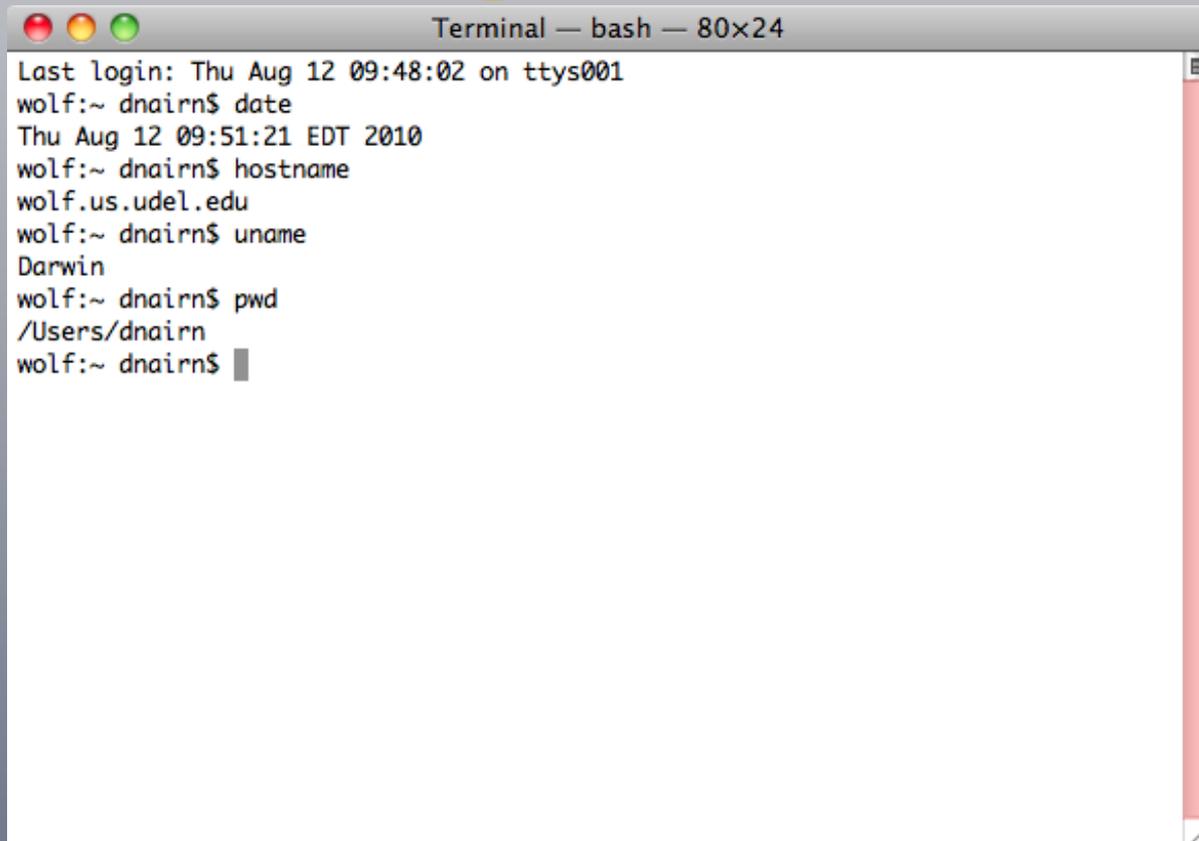


```
Terminal — bash — 80x24
Last login: Thu Aug 12 09:48:02 on ttys001
wolf:~ dnairn$ date
Thu Aug 12 09:51:21 EDT 2010
wolf:~ dnairn$ hostname
wolf.us.udel.edu
wolf:~ dnairn$ uname
Darwin
wolf:~ dnairn$ pwd
/Users/dnairn
wolf:~ dnairn$ █
```

Input  
cursor

Typed text will  
appear at the  
cursor location.

# Command Window Anatomy

A screenshot of a macOS Terminal window titled "Terminal — bash — 80x24". The window contains the following text:

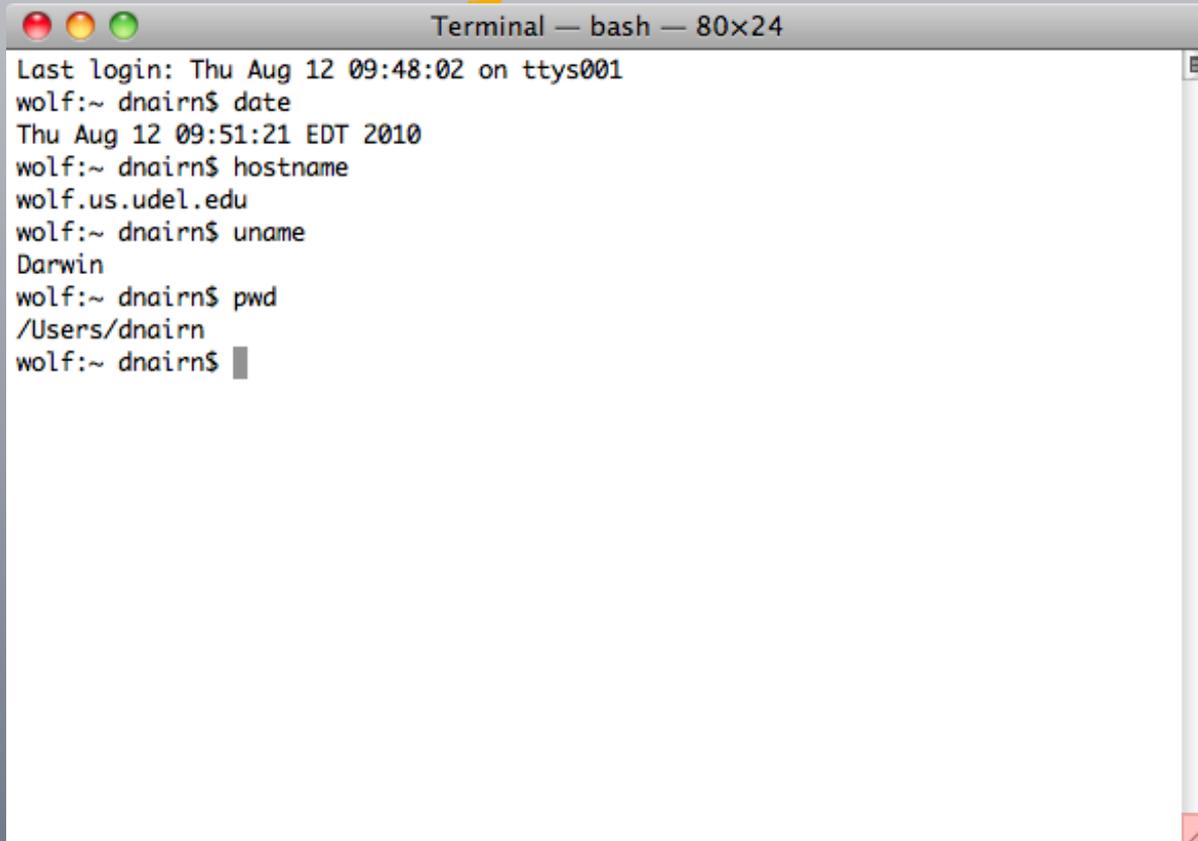
```
Last login: Thu Aug 12 09:48:02 on ttys001
wolf:~ dnairn$ date
Thu Aug 12 09:51:21 EDT 2010
wolf:~ dnairn$ hostname
wolf.us.udel.edu
wolf:~ dnairn$ uname
Darwin
wolf:~ dnairn$ pwd
/Users/dnairn
wolf:~ dnairn$ █
```

The terminal output shows the results of several system commands. A vertical scrollbar is visible on the right side of the terminal window, indicating that the content is scrollable.

Scroll Bar

Will appear as needed when there are more lines than fit in the window.

# Command Window Anatomy

A screenshot of a macOS Terminal window. The title bar reads "Terminal — bash — 80x24". The window contains the following text:

```
Last login: Thu Aug 12 09:48:02 on ttys001
wolf:~ dnairn$ date
Thu Aug 12 09:51:21 EDT 2010
wolf:~ dnairn$ hostname
wolf.us.udel.edu
wolf:~ dnairn$ uname
Darwin
wolf:~ dnairn$ pwd
/Users/dnairn
wolf:~ dnairn$
```

Resize Handle

Use the mouse to change the window size from the default 80x24.

# Command Structure

*command* [*arguments*]

- Commands are made up of the actual *command* and its *arguments*.

*command* -*options* [*arguments*]

- The arguments are further broken down into the command *options* which are single letters prefixed by a “-” and other *arguments* that identify data for the command.

# Basic Command Examples

```
train@centos:~  
[train@centos ~]$ date  
Tue Dec 14 14:11:42 EST 2010  
[train@centos ~]$ cal  
December 2010  
Su Mo Tu We Th Fr Sa  
    1  2  3  4  
  5  6  7  8  9 10 11  
12 13 14 15 16 17 18  
19 20 21 22 23 24 25  
26 27 28 29 30 31  
  
[train@centos ~]$ hostname  
centos.css.udel.edu  
[train@centos ~]$ pwd  
/home/train  
[train@centos ~]$ whoami  
train  
[train@centos ~]$ ps  
  PID TTY          TIME CMD  
19189 pts/2    00:00:00 bash  
19217 pts/2    00:00:00 ps  
[train@centos ~]$ uptime  
14:12:01 up 7 days, 17 min,  3 users,  load average: 0.00, 0.00, 0.00  
[train@centos ~]$
```

# Advanced Command Examples

```
train@centos:~  
[train@centos ~]$ history  
 1 date  
 2 cal  
 3 hostname  
 4 pwd  
 5 whoami  
 6 ps  
 7 uptime  
 8 history  
[train@centos ~]$ cal 10 2010  
  October 2010  
Su Mo Tu We Th Fr Sa  
          1  2  
 3  4  5  6  7  8  9  
10 11 12 13 14 15 16  
17 18 19 20 21 22 23  
24 25 26 27 28 29 30  
31  
[train@centos ~]$ uname  
Linux  
[train@centos ~]$ uname -a  
Linux centos.css.udel.edu 2.6.18-194.26.1.el5 #1 SMP Tue Nov 9 12:54:20 EST 2010  
x86_64 x86_64 x86_64 GNU/Linux  
[train@centos ~]$
```

# Command Input Keys

## Summary

At the cursor location use



to insert a character.



to enter the command.



to erase character to left.



to move to the left or right.



to retrieve previous command.



for command line completion.

# Command Input Keys Summary

Control **C** or **CTRL-c** or just **C-c**  
use the control key as a shift

Control **U** or **CTRL-u** or just **C-u**  
use the control key as a shift

Control **D** or **CTRL-d** or just **C-d**  
use the control key as a shift

# Command Output Summary

- Lines longer than the column width are wrapped

Use the mouse in the resize handle to increase the width of the screen and unwrap the lines.

- Lines are scrolled off the top and the prompt will appear at the bottom of the screen

Use the mouse in the scroll bar to see lines that have scrolled off the screen.

# Getting Help

## `man command`

- ▣ `man` is a command that formats and displays on-line manual pages for *command*.

## `info [subject]`

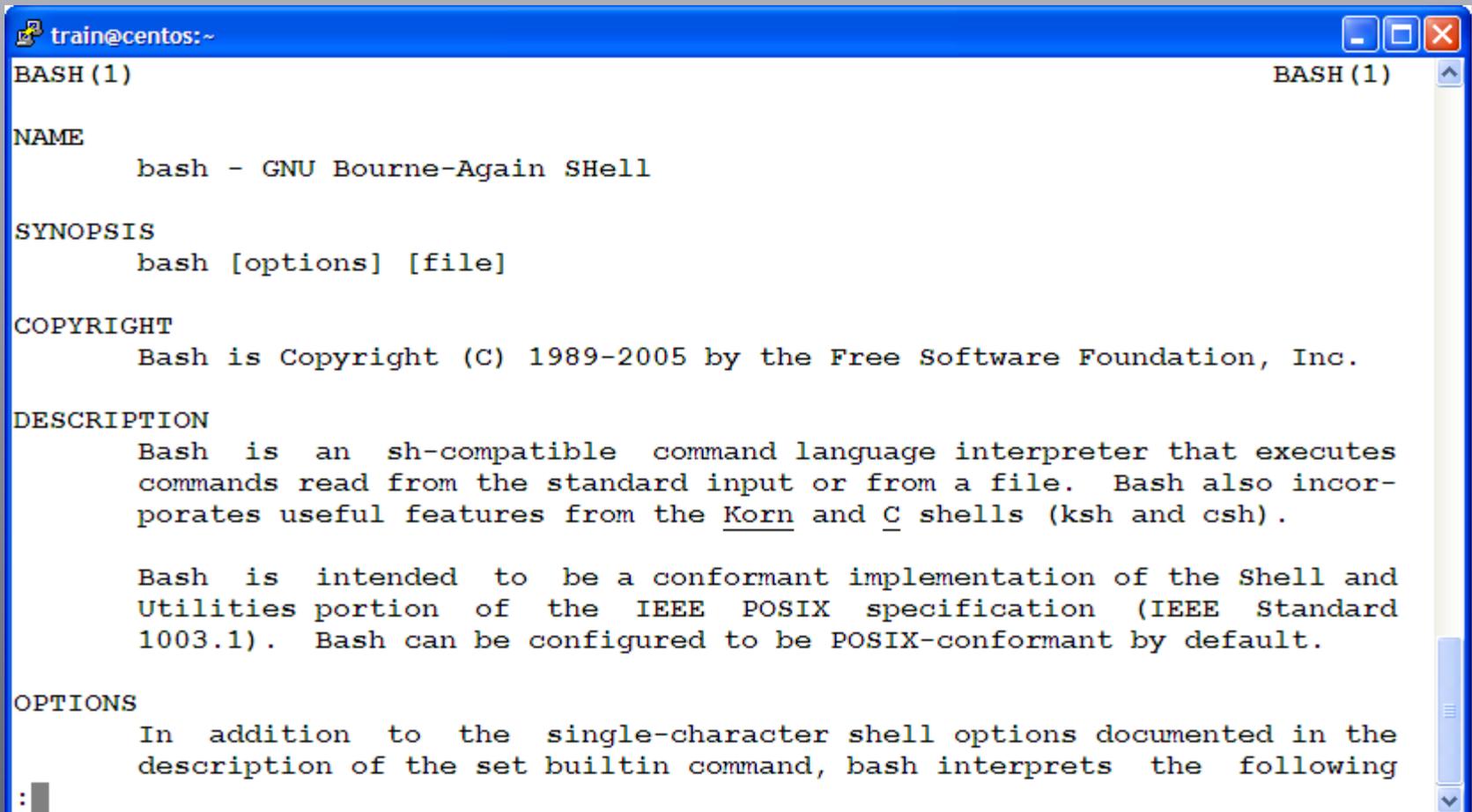
- ▣ `info` formats and displays online documents that are easily searchable.

## `apropos keyword`

- ▣ `apropos` searches the `whatis` database for commands containing the `keyword`

# Getting Help

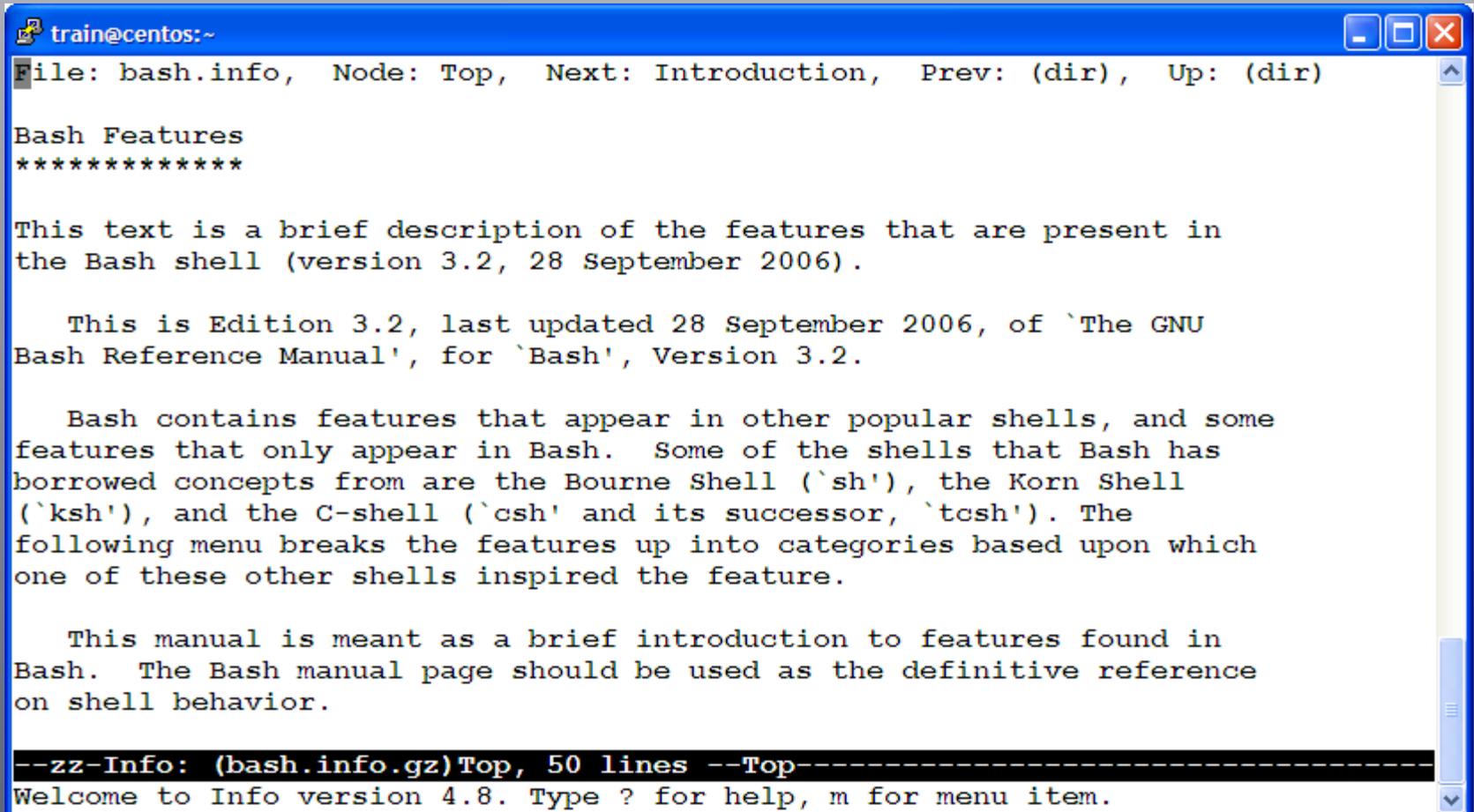
## man bash



```
train@centos:~  
BASH (1) BASH (1)  
NAME  
    bash - GNU Bourne-Again SHell  
SYNOPSIS  
    bash [options] [file]  
COPYRIGHT  
    Bash is Copyright (C) 1989-2005 by the Free Software Foundation, Inc.  
DESCRIPTION  
    Bash is an sh-compatible command language interpreter that executes  
    commands read from the standard input or from a file. Bash also incor-  
    porates useful features from the Korn and C shells (ksh and csh).  
  
    Bash is intended to be a conformant implementation of the Shell and  
    Utilities portion of the IEEE POSIX specification (IEEE Standard  
    1003.1). Bash can be configured to be POSIX-conformant by default.  
OPTIONS  
    In addition to the single-character shell options documented in the  
    description of the set builtin command, bash interprets the following  
:  
:
```

# Getting Help

## □ info bash



```
train@centos:~  
file: bash.info, Node: Top, Next: Introduction, Prev: (dir), Up: (dir)  
Bash Features  
*****  
  
This text is a brief description of the features that are present in  
the Bash shell (version 3.2, 28 September 2006).  
  
This is Edition 3.2, last updated 28 September 2006, of `The GNU  
Bash Reference Manual', for `Bash', Version 3.2.  
  
Bash contains features that appear in other popular shells, and some  
features that only appear in Bash. Some of the shells that Bash has  
borrowed concepts from are the Bourne Shell (`sh'), the Korn Shell  
(`ksh'), and the C-shell (`csh' and its successor, `tcsh'). The  
following menu breaks the features up into categories based upon which  
one of these other shells inspired the feature.  
  
This manual is meant as a brief introduction to features found in  
Bash. The Bash manual page should be used as the definitive reference  
on shell behavior.  
  
--zz-Info: (bash.info.gz)Top, 50 lines --Top-----  
Welcome to Info version 4.8. Type ? for help, m for menu item.
```

# Getting Help: Info Keys



To read linearly through all sections

delete

back up after SPC

H

'h' special link to help.

L

'l' return from help link.

Q

'q' to quit.

# Getting Help: Info Keys

tab

Jump to next link or cross reference.

return

follow the link at the cursor.

esc

tab

back up to previous link.

M

'm' to go to the menu.

L

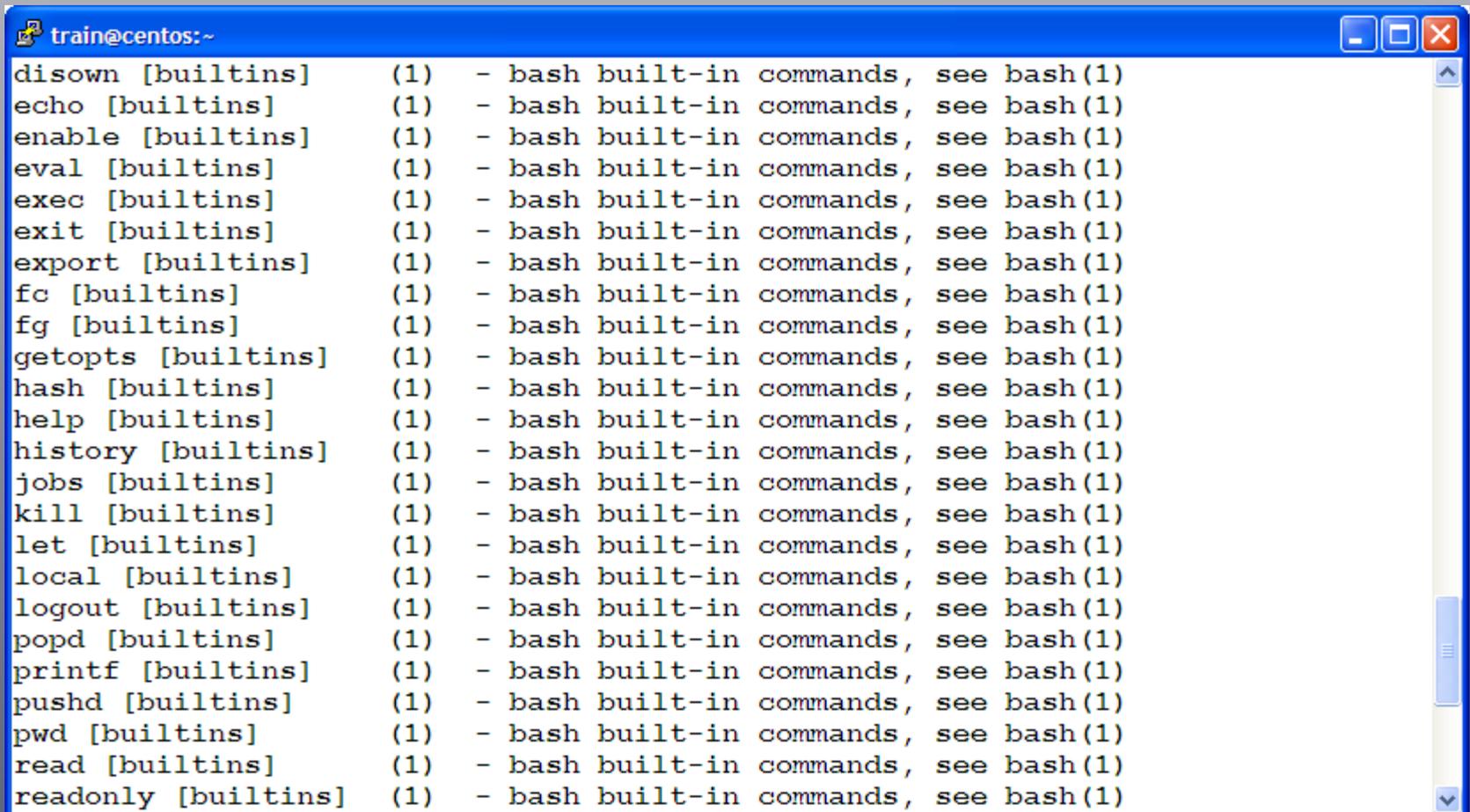
'l' return from link.

Q

'q' to quit.

# Getting Help

## ▮ apropos bash



```
train@centos:~  
disown [builtins]      (1) - bash built-in commands, see bash(1)  
echo [builtins]       (1) - bash built-in commands, see bash(1)  
enable [builtins]    (1) - bash built-in commands, see bash(1)  
eval [builtins]      (1) - bash built-in commands, see bash(1)  
exec [builtins]      (1) - bash built-in commands, see bash(1)  
exit [builtins]      (1) - bash built-in commands, see bash(1)  
export [builtins]    (1) - bash built-in commands, see bash(1)  
fc [builtins]        (1) - bash built-in commands, see bash(1)  
fg [builtins]        (1) - bash built-in commands, see bash(1)  
getopts [builtins]  (1) - bash built-in commands, see bash(1)  
hash [builtins]     (1) - bash built-in commands, see bash(1)  
help [builtins]     (1) - bash built-in commands, see bash(1)  
history [builtins]  (1) - bash built-in commands, see bash(1)  
jobs [builtins]     (1) - bash built-in commands, see bash(1)  
kill [builtins]     (1) - bash built-in commands, see bash(1)  
let [builtins]      (1) - bash built-in commands, see bash(1)  
local [builtins]    (1) - bash built-in commands, see bash(1)  
logout [builtins]  (1) - bash built-in commands, see bash(1)  
popd [builtins]    (1) - bash built-in commands, see bash(1)  
printf [builtins]  (1) - bash built-in commands, see bash(1)  
pushd [builtins]   (1) - bash built-in commands, see bash(1)  
pwd [builtins]     (1) - bash built-in commands, see bash(1)  
read [builtins]    (1) - bash built-in commands, see bash(1)  
readonly [builtins] (1) - bash built-in commands, see bash(1)
```

# Files and Directories

## Files

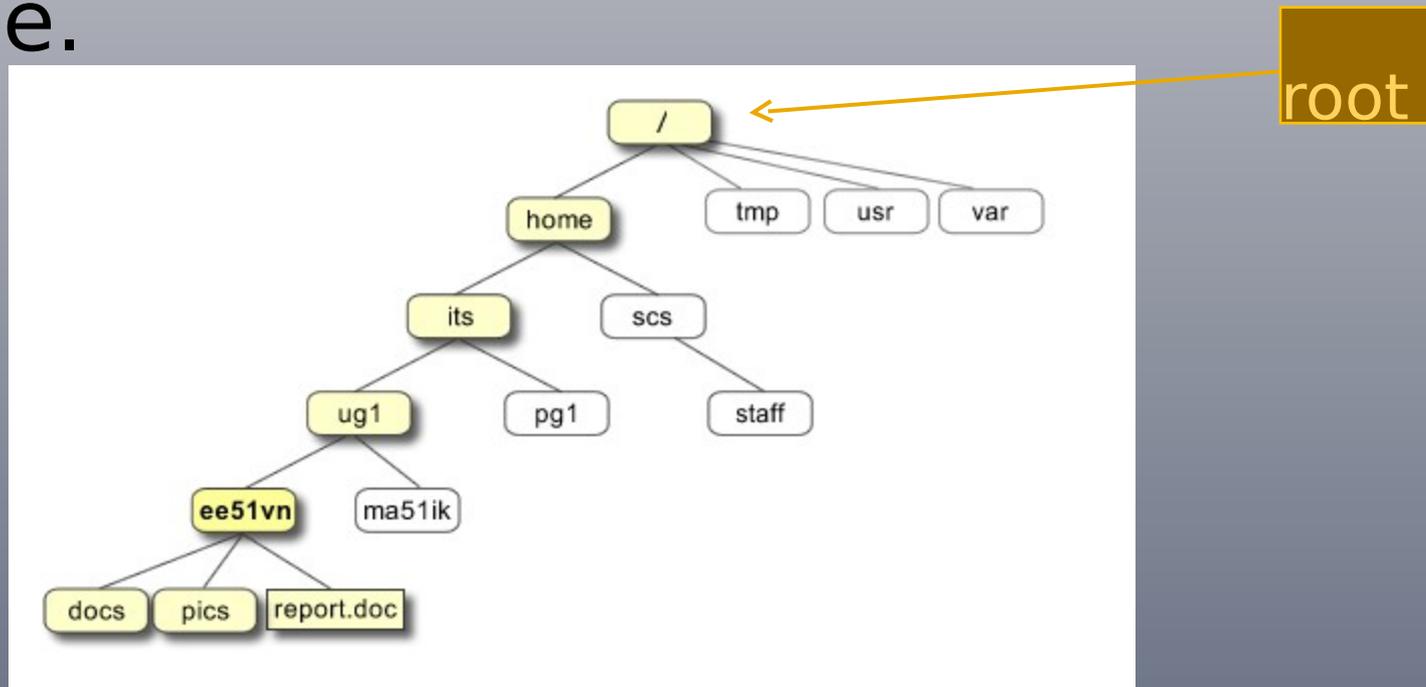
- Data in UNIX is stored in files.

## Directories

- Files are organized hierarchically into directories.
- Top-level directory is called “root” represented as a slash (/).
- Every file (and directory) is listed and separated by a slash (/).

# Files and Directories

The UNIX file system is like an inverted tree.



/  
home/its/ug1/ee51vn/report.

# Working with Files

```
train@centos:~  
[train@centos ~]$ ls  
bessel.c  besselTest  birthtoken.txt  example.c  flowers.txt  lines  
[train@centos ~]$ ls -a  
.          .bash_logout  bessel.c        example.c    lines  
..         .bash_profile besselTest      flowers.txt  .mozilla  
.bash_history .bashrc       birthtoken.txt  .lessht     .Xauthority  
[train@centos ~]$ ls -la  
total 120  
drwxr-xr-x  4 train      508 4096 Dec 14 14:30 .  
drwxr-xr-x 10 root      root  4096 Dec  2 19:27 ..  
-rw-----  1 train  student   91 Dec 14 14:11 .bash_history  
-rw-r--r--  1 train  student   33 Dec  7 11:11 .bash_logout  
-rw-r--r--  1 train  student  176 Dec  7 11:11 .bash_profile  
-rw-r--r--  1 train  student  124 Dec  7 11:11 .bashrc  
-rw-r-----  1 train  student  173 Dec 14 14:30 bessel.c  
drwxr-x---  2 train  student 4096 Dec 14 14:30 besselTest  
-r--r-----  1 train  student  374 Dec 14 14:30 birthtoken.txt  
-rw-r-----  1 train  student  173 Dec 14 14:30 example.c  
-r--r-----  1 train  student 1391 Dec 14 14:30 flowers.txt  
-rw-----  1 train  student   44 Dec 14 14:16 .lessht  
-rw-r-----  1 train  student 1492 Dec 14 14:30 lines  
drwxr-xr-x  4 train  student 4096 Dec  7 11:11 .mozilla  
-rw-----  1 train  student  130 Dec 14 14:11 .Xauthority  
[train@centos ~]$
```

# Working with Files

```
train@centos:~  
[train@centos ~]$ touch mytest  
[train@centos ~]$ ls  
bessel.c  besselTest  birthtoken.txt  example.c  flowers.txt  lines  mytest  
[train@centos ~]$ cp mytest mytest.new  
[train@centos ~]$ ls  
bessel.c  birthtoken.txt  flowers.txt  mytest  
besselTest  example.c      lines      mytest.new  
[train@centos ~]$ rm mytest  
[train@centos ~]$ ls  
bessel.c  besselTest  birthtoken.txt  example.c  flowers.txt  lines  mytest.new  
[train@centos ~]$ cp mytest.new mytest  
[train@centos ~]$ ls  
bessel.c  birthtoken.txt  flowers.txt  mytest  
besselTest  example.c      lines      mytest.new  
[train@centos ~]$ mv mytest.new mytest.old  
[train@centos ~]$ ls  
bessel.c  birthtoken.txt  flowers.txt  mytest  
besselTest  example.c      lines      mytest.old  
[train@centos ~]$ rm -i mytest.old  
rm: remove regular empty file `mytest.old'? yes  
[train@centos ~]$ ls  
bessel.c  besselTest  birthtoken.txt  example.c  flowers.txt  lines  mytest  
[train@centos ~]$  
[train@centos ~]$
```

# Working with Directories

```
train@centos:~  
[train@centos ~]$ pwd  
/home/train  
[train@centos ~]$ ls  
bessel.c  besselTest  birthtoken.txt  example.c  flowers.txt  lines  mytest  
[train@centos ~]$ mkdir src  
[train@centos ~]$ ls  
bessel.c  birthtoken.txt  flowers.txt  mytest  
besselTest  example.c  lines  src  
[train@centos ~]$ cd src  
[train@centos src]$ mkdir C  
[train@centos src]$ mkdir TXT  
[train@centos src]$ ls  
C  TXT  
[train@centos src]$ ls -la  
total 32  
drwxr-xr-x 4 train student 4096 Dec 14 14:45 .  
drwxr-xr-x 5 train student 4096 Dec 14 14:45 ..  
drwxr-xr-x 2 train student 4096 Dec 14 14:45 C  
drwxr-xr-x 2 train student 4096 Dec 14 14:45 TXT  
[train@centos src]$ cd  
[train@centos ~]$ pwd  
/home/train  
[train@centos ~]$
```

# Working with Directories:

## Home (~) and Wildcards

```
train@centos:~/src/TXT
[train@centos ~]$ cd src
[train@centos src]$ cd C
[train@centos C]$ ls
[train@centos C]$ ls ~
bessel.c      birthtoken.txt  flowers.txt    mytest
besselTest   example.c       lines          src
[train@centos C]$ pwd
/home/train/src/C
[train@centos C]$ cp ~/*.c .
[train@centos C]$ ls
bessel.c  example.c
[train@centos C]$ ls ~
bessel.c      birthtoken.txt  flowers.txt    mytest
besselTest   example.c       lines          src
[train@centos C]$ pwd
/home/train/src/C
[train@centos C]$ cd ..
[train@centos src]$ ls
C  TXT
[train@centos src]$ cd TXT
[train@centos TXT]$ cp ~/*.txt .
[train@centos TXT]$ ls
birthtoken.txt  flowers.txt
[train@centos TXT]$
```

# Working with Directories

```
train@centos:~/src
[train@centos ~]$ cd
[train@centos ~]$ ls
bessel.c  besselTest  birthtoken.txt  example.c  flowers.txt  lines  src
[train@centos ~]$ rm src
rm: cannot remove `src': Is a directory
[train@centos ~]$ cd src
[train@centos src]$ ls
C  TXT
[train@centos src]$ ls C
bessel.c  example.c
[train@centos src]$ ls TXT
birthtoken.txt  flowers.txt
[train@centos src]$ rmdir C
rmdir: C: Directory not empty
[train@centos src]$ mv C CC
[train@centos src]$ ls
CC  TXT
[train@centos src]$ ls CC
bessel.c  example.c
[train@centos src]$ rm CC/bessel.c
[train@centos src]$ ls CC
example.c
[train@centos src]$
```

# Working with Directories and Files:

```
train@centos:~  
[train@centos src]$ rm -ri CC  
rm: descend into directory `CC'? y  
rm: remove regular file `CC/example.c'? n  
[train@centos src]$ rm -r CC  
[train@centos src]$ ls  
TXT  
[train@centos src]$ rm -r TXT  
rm: remove write-protected regular file `TXT/birthtoken.txt'? y  
rm: remove write-protected regular file `TXT/flowers.txt'? y  
[train@centos src]$ ls  
[train@centos src]$ pwd  
/home/train/src  
[train@centos src]$ cd  
[train@centos ~]$ ls -lad .??*  
-rw----- 1 train student 3622 Jan  7 17:00 .bash_history  
-rw-r--r-- 1 train student   33 Dec  7 11:11 .bash_logout  
-rw-r--r-- 1 train student  176 Dec  7 11:11 .bash_profile  
-rw-r--r-- 1 train student  124 Dec  7 11:11 .bashrc  
drwx----- 2 train student 4096 Jan  6 15:03 .gconf  
drwx----- 2 train student 4096 Jan  6 15:42 .gconfd  
drwx----- 3 train student 4096 Jan  6 14:57 .gnome2  
drwx----- 2 train student 4096 Jan  6 14:57 .gnome2_private  
-rw----- 1 train student   83 Jan  6 01:15 .lessht  
drwxr-xr-x 5 train student 4096 Jan  2 23:15 .mozilla  
-rw----- 1 train student 3838 Jan  7 15:57 .viminfo  
-rw----- 1 train student  455 Jan 10 06:28 .Xauthority  
[train@centos ~]$ ls *.*  
bessel.c example.c  
[train@centos ~]$
```

# Creating Files: Redirection (STDOUT)

```
train@centos:~  
[train@centos ~]$ cat > animals-sm.txt  
mouse  
finch  
hamster  
[train@centos ~]$ cat > animals-lg.txt  
elephant  
buffalo  
rhinoceros  
[train@centos ~]$ more animals*  
:::::::::::::  
animals-lg.txt  
:::::::::::::  
elephant  
buffalo  
rhinoceros  
:::::::::::::  
animals-sm.txt  
:::::::::::::  
mouse  
finch  
hamster  
[train@centos ~]$
```

^D (control d for EOF)

# Creating Files: Redirection

```
train@centos:~  
[train@centos ~]$ cat animals-sm.txt animals-lg.txt > animals.txt  
[train@centos ~]$ more animals.txt  
mouse  
finch  
hamster  
elephant  
buffalo  
rhinoceros  
[train@centos ~]$ sort animals.txt  
buffalo  
elephant  
finch  
hamster  
mouse  
rhinoceros  
[train@centos ~]$ sort animals.txt > animals-sorted.txt  
[train@centos ~]$ more animals-sorted.txt  
buffalo  
elephant  
finch  
hamster  
mouse  
rhinoceros  
[train@centos ~]$ █
```

# Creating Files: Append and Pipe

```
train@centos:~  
[train@centos ~]$ cat >> animals-lg.txt  
cow  
horse  
[train@centos ~]$ more animals-lg.txt  
elephant  
buffalo  
rhinoceros  
cow  
horse  
[train@centos ~]$ cat animals-lg.txt animals-sm.txt | sort > animals-sorted.txt  
[train@centos ~]$ ls  
animals-lg.txt      animals.txt  birthtoken.txt  lines  
animals-sm.txt      bessel.c   example.c       mytest  
animals-sorted.txt  besselTest flowers.txt      src  
[train@centos ~]$ more animals-sorted.txt  
buffalo  
cow  
elephant  
finch  
hamster  
horse  
mouse  
rhinoceros  
[train@centos ~]$
```

**^D (control d to stop)**

# Creating Files: Redirection (STDIN) and

```
train@centos:~  
[train@centos ~]$ more animals.txt  
mouse  
finch  
hamster  
elephant  
buffalo  
rhinoceros  
[train@centos ~]$ sort < animals.txt > animals-orig-sort.txt  
[train@centos ~]$ more animals-orig-sort.txt  
buffalo  
elephant  
finch  
hamster  
mouse  
rhinoceros  
[train@centos ~]$ cat flowers.txt birthtoken.txt | grep -i violet | sort  
African violet:Such worth is rare.  
February:Amethyst:Violet  
Violet, blue:Faithfulness.  
Violet, white:Modesty.  
[train@centos ~]$ cat flowers.txt birthtoken.txt | grep -i violet | sort | wc -l  
4  
[train@centos ~]$ █
```

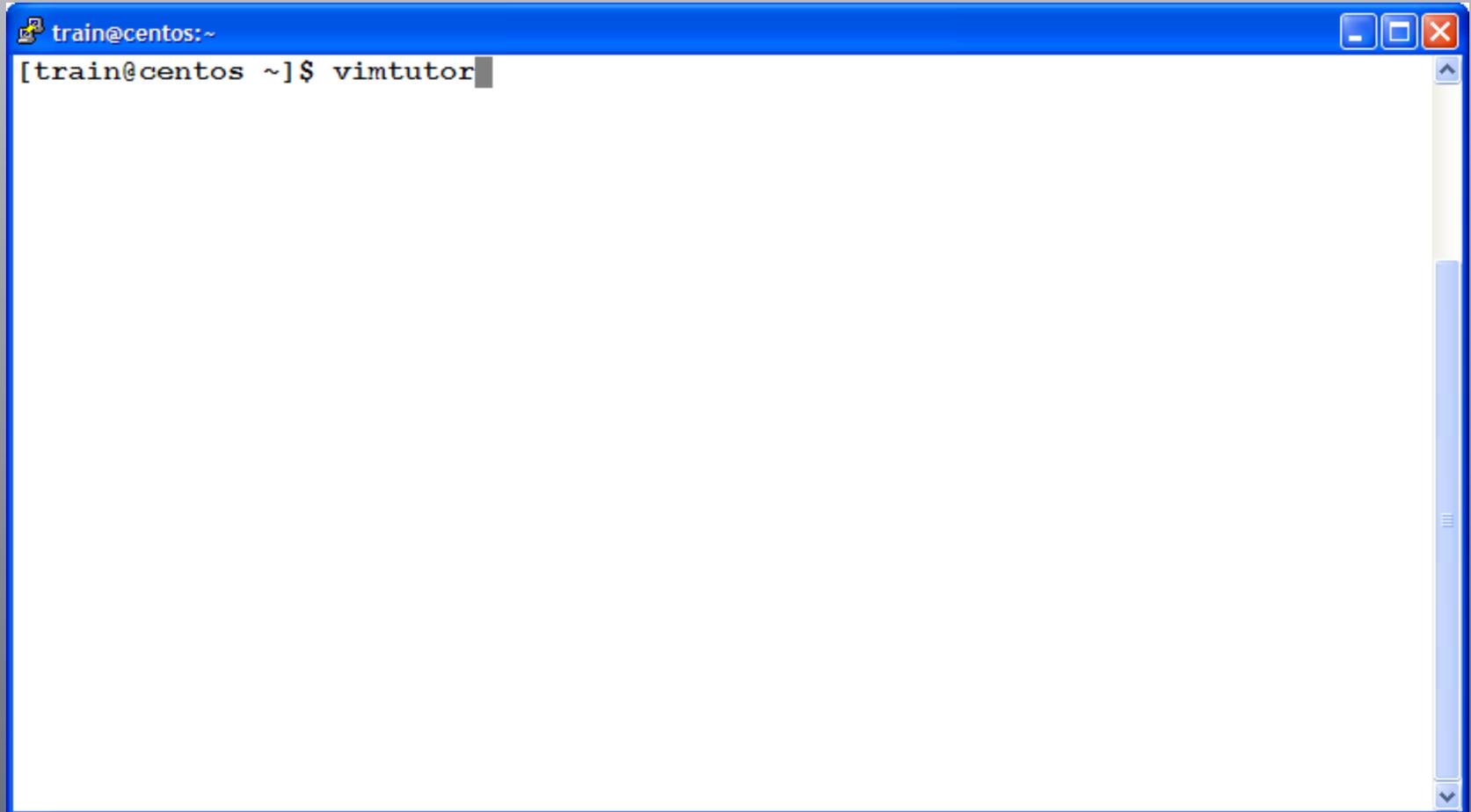
# Editing Files

The VIM editor has two modes:

- 1. Command:** interprets a letter or sequence of letters as a command.
- 2. Insert:** puts anything typed into the file. The ESC key ends insert mode and returns you to command mode.

**Command line** entry at the bottom of the screen appears when the command “:” is typed. VIM starts up in command mode.

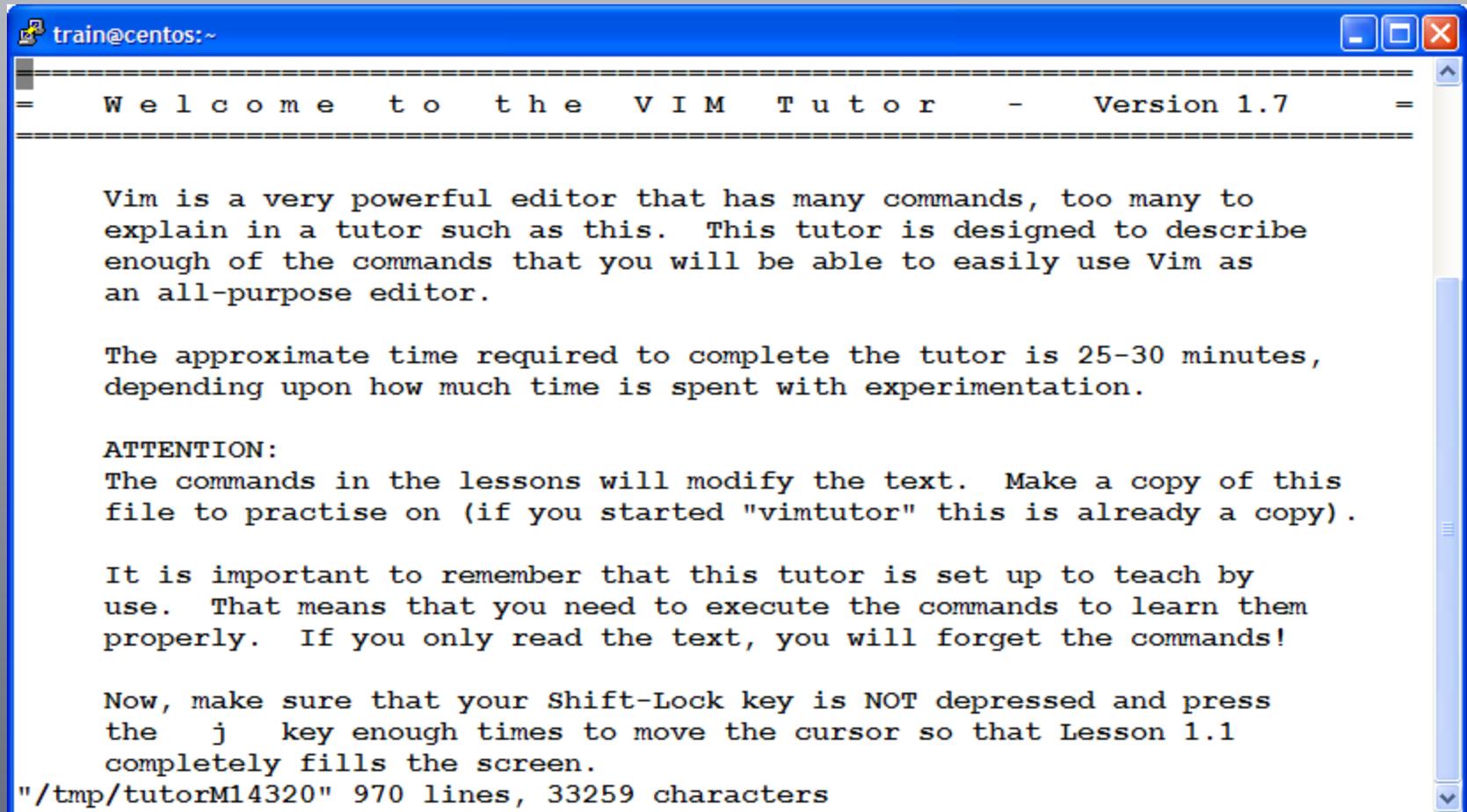
# VIM Tutor: Getting Started



A terminal window with a blue title bar. The title bar contains the text 'train@centos:~' and standard window control buttons (minimize, maximize, close). The terminal content shows the prompt '[train@centos ~]\$' followed by the command 'vimtutor' and a cursor. The rest of the terminal area is empty.

```
train@centos:~  
[train@centos ~]$ vimtutor
```

# VIM tutor: Getting Started

A screenshot of a terminal window titled "train@centos:~". The window displays the VIM tutor interface. At the top, a title bar shows the window title and standard Linux window controls (minimize, maximize, close). Below the title bar, a separator line is followed by the text "Welcome to the VIM Tutor - Version 1.7" centered on the screen. Another separator line follows. The main content consists of several paragraphs of text explaining the purpose of the tutor, the time required to complete it, and instructions for using the tutor. The text is formatted in a monospaced font. At the bottom of the window, a status line shows the file path and statistics: "/tmp/tutorM14320" 970 lines, 33259 characters. The window also features a vertical scrollbar on the right side.

```
train@centos:~  
===== Welcome to the VIM Tutor - Version 1.7 =====  
  
Vim is a very powerful editor that has many commands, too many to  
explain in a tutor such as this. This tutor is designed to describe  
enough of the commands that you will be able to easily use Vim as  
an all-purpose editor.  
  
The approximate time required to complete the tutor is 25-30 minutes,  
depending upon how much time is spent with experimentation.  
  
ATTENTION:  
The commands in the lessons will modify the text. Make a copy of this  
file to practise on (if you started "vimtutor" this is already a copy).  
  
It is important to remember that this tutor is set up to teach by  
use. That means that you need to execute the commands to learn them  
properly. If you only read the text, you will forget the commands!  
  
Now, make sure that your Shift-Lock key is NOT depressed and press  
the j key enough times to move the cursor so that Lesson 1.1  
completely fills the screen.  
"/tmp/tutorM14320" 970 lines, 33259 characters
```

# Resources

- Unix Tutorial for Beginners  
<http://info.ee.surrey.ac.uk/Teaching/Unix/>
- VTC (Unix Shell Fundamentals) – need to request an account  
<http://www.udel.edu/it/learnit/course/vtccom.html>
- VIM Tutor (vimtutor)
- Linux vi and vim editor: Tutorial and advanced features  
[http://www.yolinux.com/TUTORIALS/LinuxTutorialAdvanced\\_vi.html](http://www.yolinux.com/TUTORIALS/LinuxTutorialAdvanced_vi.html)
- Graphical vi-vim Cheat Sheet and Tutorial  
[http://www.viemu.com/a\\_vi\\_vim\\_graphical\\_cheat\\_sheet\\_tutorial.html](http://www.viemu.com/a_vi_vim_graphical_cheat_sheet_tutorial.html)