

# Introduction to the Command Line

Information Services & Technology

# The Shell

- Interface between user and operating system (OS)
  - Interprets what you type and tells the OS what to do
- Two primary types of shells
  - Bourne shell and C shell
  - "bash" and "tcsh"
  - All Athena accounts since 4/1/09 use bash

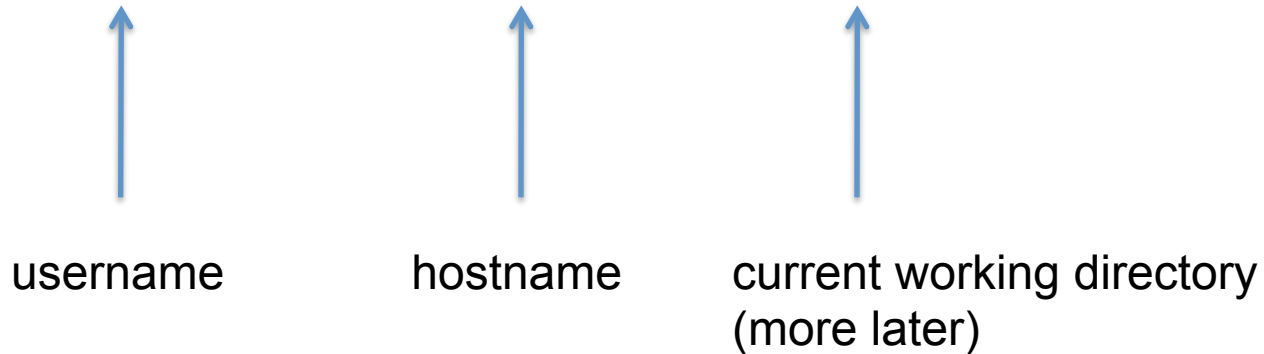
# Common Shells

- Bourne-shell family
  - /bin/sh, /bin/bash
- C-Shell family
  - /bin/csh, /bin/tcsh
- DOS
  - COMMAND.COM
- Windows
  - cmd.exe
- MATLAB
  - >>

# Shell prompt

- Tells you the shell is ready for your input
- Bourne shell prompts end with dollar sign (C shell prompts end with `>` or `%`)

– e.g. `joeuser@w20-575-1:~$`

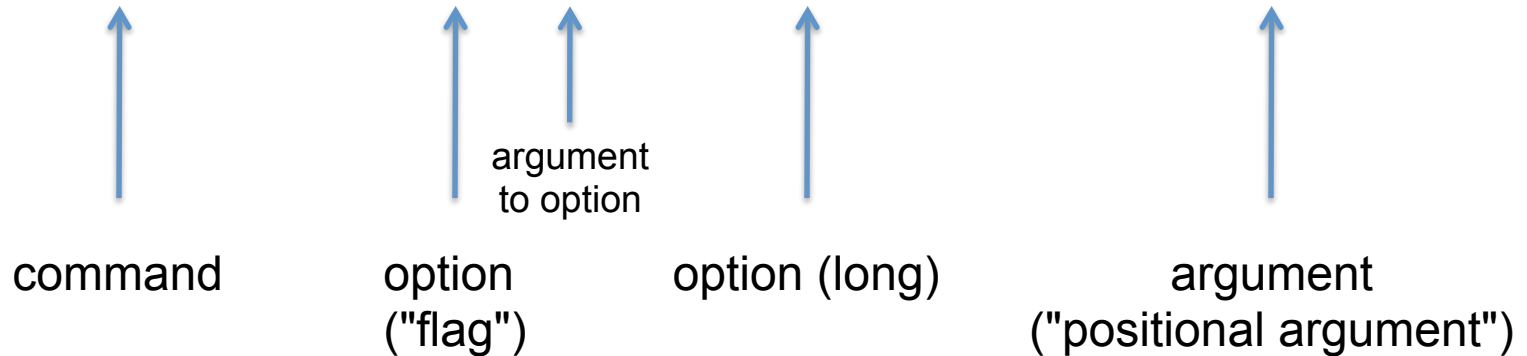


# Filenames

- Filenames are case-sensitive
  - EXAM.DOC != exam.doc
- Spaces and other special characters can be used too, but you need to surround the filename with quotation marks
  - "my special file.doc"
  - "foo&bar.doc"

# Commands

- `dothis -c 1 --special paper.doc`



- command can take 0 or more arguments and 0 or more options
- options themselves can take arguments  
e.g. `--copies 5` or `--copies=5`

# Stopping a command

- Ctrl-C
  - Returns you to your shell
- Useful when a command is expecting input, or running too long
- Or if you're not at all sure what you just did

# ls - list contents of directory

- ls

```
joeuser@mint-square:~$ ls
```

```
Desktop      Downloads  Pictures   Templates  welcome
Documents   Music      Public     Videos
```

- ls -l ("long" listing)

```
joeuser@mint-square:~$ ls -l
```

```
drwxr-xr-x 2 joeuser nogroup 2048 2011-08-01 17:58 Desktop
drwxr-xr-x 3 joeuser nogroup 2048 2011-09-08 14:40 Documents
drwxr-xr-x 2 joeuser nogroup 2048 2010-08-24 14:34 Downloads
drwxr-xr-x 2 joeuser nogroup 2048 2010-08-24 14:34 Music
drwxr-xr-x 3 joeuser nogroup 2048 2011-01-05 14:12 Pictures
drwxr-xr-x 2 joeuser nogroup 2048 2011-08-09 12:01 Public
drwxr-xr-x 2 joeuser nogroup 2048 2010-08-24 14:34 Templates
drwxr-xr-x 3 joeuser nogroup 2048 2011-01-05 14:12 Videos
-rw-r--r-- 1 joeuser nogroup 1819 2011-09-08 16:07 welcome
```



# ls -l

```
drwxr-xr-x 2 joeuser nogroup 2048 2011-08-01 17:58 Desktop
```



mode bits

owner

group

size

date/time

name

- owner and group don't mean much in AFS (more later)
- date/time is last modification
- mode bits
  - first column: type of file ('d' = directory '-' = regular file, others)
  - next 9 columns divided into 3 groups
    - access rights for "owner" "group" and "other" (everyone else)
    - "group" and "other" not used in AFS

# cat

- cat

- Display the contents of a file

```
joeuser@mint-square:~$ cat welcome
```

```
Welcome to Athena
```

```
-----
```

```
- Project Athena was an eight-year experiment (1983-1991) to explore the use  
  of computers in university educational programs. The Project focused on  
  [...]
```

# head and tail

- head and tail allow you to see the beginning or end of a file
  - by default, the first 10 or last 10 lines
- Pass numeric options for the number of lines you want

```
joeuser@mint-square:~$ head -2 welcome
Welcome to Athena
-----
```

- tail is useful for log files

# Pagers

- Display one "page" (screen) of information at a time
- more
  - `more welcome`
  - (press Space to advance a page, Enter to advance a line)
- less
  - more advanced features, go backwards, edit

# Directories ("Folders")

- **mkdir** - create a directory
  - `mkdir psets`
  - `mkdir psets/18.03`
- **rmdir** - remove a directory
  - only when it's empty
  - `rmdir psets`

# Copying, Moving and Renaming

- **cp - copy files**

- `cp thesis.doc thesis-copy.doc`

- make a copy of a file

- `cp group-work.doc Public`

- copy a file into a directory (directory must exist)

- **mv - move or rename files**

- `mv paper.doc super-awesome-paper.doc`

- rename a file

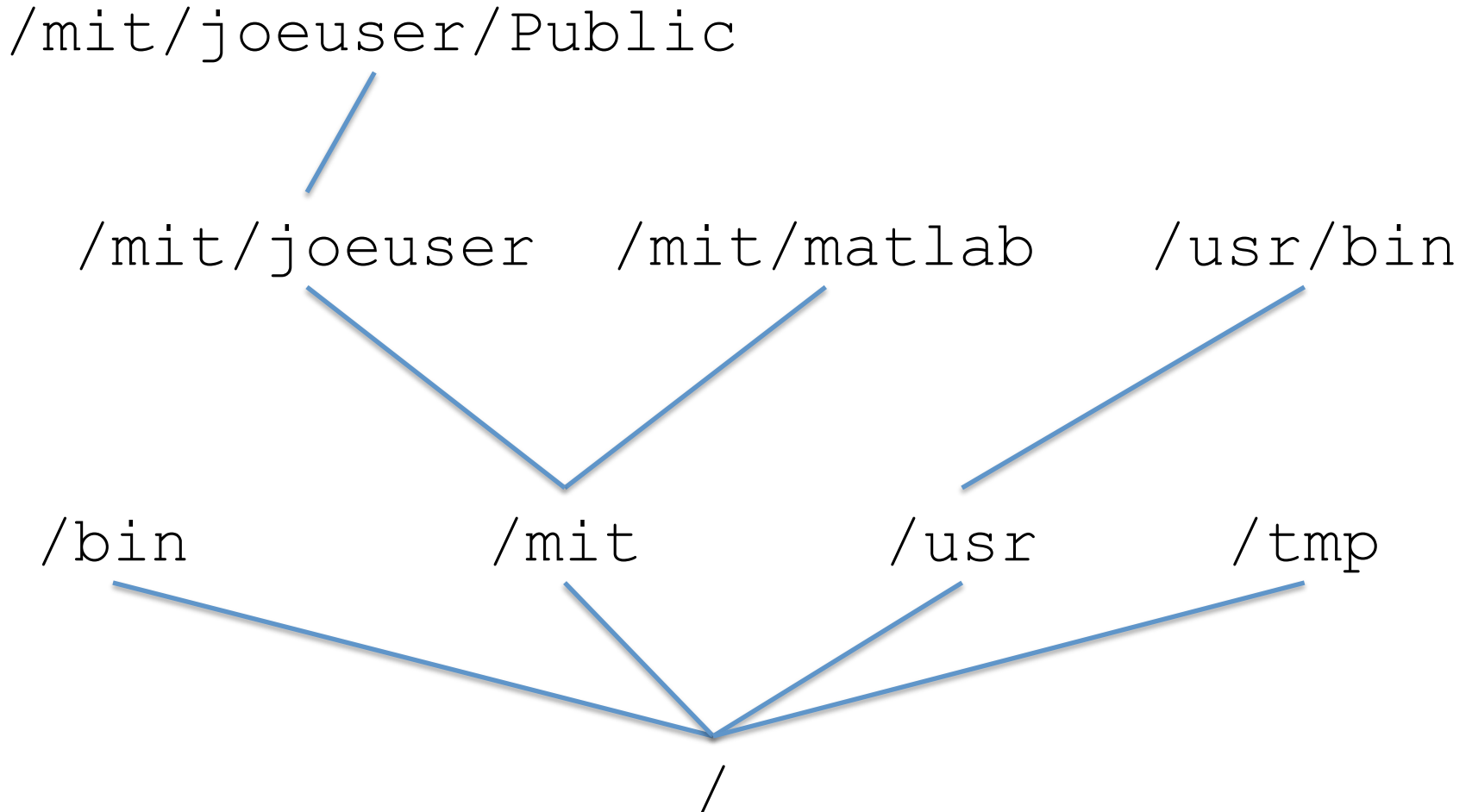
- `mv paper.doc Documents`

- move file into directory (must exist)

# Deleting files

- **rm**
  - "rm is forever"
- **delete**
  - `delete junk.doc`
  - `delete thesis.doc`
  - `lsdel`
  - `undelete thesis.doc`
  - `expunge`

# The filesystem





# The filesystem

- All directories under /
- There are no "drives" as in the DOS/Windows world
  - Different locations under / can be on the same physical disk or different disks or remote servers

# What else is in the tree?

- Defined by the "Filesystem Hierarchy Standard"
  - <http://www.pathname.com/fhs/>
- /mit - lockers
- /afs - AFS "cells"
- /tmp - temporary storage
- /bin, /usr/bin - programs ("binaries")

# Navigating the tree

- Where you are at any time is called the "working directory"
  - pwd = print working directory

```
joeuser@mint-square:~$ pwd
/afs/athena.mit.edu/user/j/o/joeuser
```

    - Why not /mit/joeuser? Stay tuned...
- To change a directory, use cd
  - cd /tmp
  - cd Documents

# Relative and absolute paths

- An *absolute path* is a path that begins with /
  - `cd /mit/joeuser`
- A *relative path* does not begin with /
  - `cd Documents`
  - It is *relative* to the current working directory
  - The command above assumes there is a directory called Documents in the current working directory
  - `cd Documents/Fall2011/21L.011`

# Special directories

- `.`
  - the current directory
- `..`
  - up one level in the tree
- `~`
  - your home directory
  - `cd ~/Public`

# Wildcards

- \* = 0 or more characters
- ? = exactly one character
- `thesis*` matches
  - `thesis`
  - `thesis.doc`
  - `thesis-backup.doc`
- `thesis.doc?` matches
  - `thesis.docx`
  - **but NOT** `thesis.doc`

# Links

- "Shortcut", "Alias"
- Special "file" that says "Go look over there"
  - "Symbolic" links simply refer to another path, which may or may not exist
  - "Hard" links actually share the same physical data on disk.
- `ls -l` will show you where they point to

# Accessing lockers

- `/mit/joeuser`
  - symbolic link to  
`/afs/athena.mit.edu/user/j/o/  
joeuser`
- "Automounter"
  - `ls /mit`
  - `cd /mit/jruser`
  - `ls /mit`



# Your PATH

- How does the shell know where the programs actually are?
  - PATH - a colon-separated list of directories to look for programs executed with a relative path
- `echo $PATH`

# add - modifying your path

- The add command adds a location to your path
  - usually a locker, but you can use it for any directory
- `add matlab`
  - Look in `/mit/matlab`, find the version of the programs that match the workstation you're using, and add them to your PATH

# AFS permissions

- Per-directory basis
- Permissions can be assigned to users or groups ("Moira lists")
  - This is why the "mode bits" in `ls` output are ignored
- 7 modes (can be combined)
  - **R**ead, **L**ist, **I**nset, **D**elete, **W**rite, **l**ock, **A**dminister
  - or the aliases `read`, `write`, `all`, `or none`

# Viewing Permissions

- `fs listacl -dir directory`
- **Can be abbreviated**
  - `fs la directory`
  - If directory omitted, defaults to current working directory
- **Example:**

```
joeuser@mint-square:~$ fs la
```

```
Access list for . is
```

```
Normal rights:
```

```
system:expunge ld
```

```
system:anyuser l
```

```
joeuser rlidwka
```

# Granting Permission

- `fs setacl -dir directory -acl entity mode`
  - *directory* - a relative or absolute path
  - *entity* - a username or `system:group`
  - *mode* - e.g. `r1` or aliases like `read`
- **Can be abbreviated:**  
`fs sa directory entity mode`
- **Example**
  - `mkdir shared`
  - `fs sa shared jruser read`

# Special AFS Entities

- `system:anyuser`
  - Anyone, anywhere, including access via the web
- `system:authuser`
  - Anyone with an Athena account
- `system:expunge`
  - Automated expunger (for 'delete')

# Moira Lists & AFS Groups

- Moira lists
  - Used for both e-mail lists and groups
  - <http://ist.mit.edu/services/email/maillists>
- Manage with <http://web.mit.edu/moira/>
- `or listmaint`
- `or blanche`

# blanche

- `blanche listname`

- **display membership**

```
joeuser@mint-square:~$ blanche myfriends
joeuser
jruser
```

- `blanche listname -i`

- **Information about a list**

```
joeuser@mint-square:~$ blanche -i myfriends
List: myfriends
Description: Reserved: this group name is used in documentation
Flags: active, private, and visible
myfriends is not a maillist and is a group with GID 73378
Owner: USER joeuser
```



# blanche

- `blanche listname -a newuser`
  - add "newuser" to the list
  - Can add other lists or external (non-MIT) e-mail addresses
    - non-MIT users won't be able to access your files
- `blanche listname -d olduser`
  - Remove "olduser" from the list
- `blanche listname -G`
  - Make the list a group (if it wasn't already)

# Your Disk Quota

- 2GB
  - Ask `accounts@mit.edu` if you want more
- Viewing your quota usage
  - `quota -v`
  - `fs lq ~`
- If you go over your quota, you can't log in

# semicolon and backslash

- Separate multiple commands on the same line with semicolons
  - `cd Public ; ls`
- The backslash (`\`) is used to "escape" characters that would otherwise be interpreted by the shell
  - `cp my paper.doc Documents`
    - tries to copy "my" and "paper.doc" to Documents
  - `cp my\ paper.doc Documents`
    - copies "my paper.doc" to Documents

# Tab Completion

- Avoids you having to type full pathnames
  - `cd Public`
  - `cd Pu<press Tab>`
- If the first few letters you type are ambiguous (could refer to multiple directories), press Tab twice to show what your options are
  - `cd P<tab><tab>`

# History and Line Editing

- Use up/down arrows to scroll through previous commands
  - Edit them, fix typos, etc
- use `history` command
  - re-run previous commands with exclamation point and job number (from history output)
  - `!501`
  - `!-1`
    - Run previous command

# Manual Pages

- Online "manual"
- `man ls`
- Synopsis: use of command  
Description: describes options  
See Also: cross references to other pages
- `command(section)`
  - `lpadmin(8)`
  - `man 8 lpadmin`

# Input and Output (I/O)

- How does it know to get input from keyboard?  
How does it know to print results and errors to screen?
- Standard Input (stdin)
- Standard Output (stdout)
- Standard Error (stderr)
- File descriptors 0, 1 and 2 respectively
  - File descriptor is unique number used by OS to refer to file (or file-like device)

# I/O redirection

- `ls Public > myfiles.txt`
- `ls Pubblic > myfiles.txt`  
`ls: cannot access Pubblic: No such file or directory`
  - Only redirected stdout, not stderr
- `ls Public > myfiles.txt 2>&1`
  - send stdout to output.txt
  - send fd 2 (stderr) to fd 1 (stdout)
- `rm file-that-might-exist.doc 2>/dev/null`



# Input Redirection and Pipes

- Input redirection
  - `zwrite joeuser < mymessage.txt`
- Pipes - connect stdout of one command to stdin of another
  - `cat very-long-file.txt | more`
  - `sed -e 's/Harvard/MIT/g' < thesis.txt > plagiarism.txt`
- Use `| &` to redirect stdout and stderr to the new command

# Foreground, Background

- Normally, jobs run in foreground
  - Can't do anything while a long command is running
- Run commands in background with ampersand
  - `xterm &`
  - get shell prompt back immediately to run other commands

# Job Management

- For an already-running command
  - Suspend (stop) it with Ctrl-Z
  - background it with `bg`
- the `jobs` command displays backgrounded and stopped jobs
  - bring them to foreground with `fg`
- `fg` and `bg` operate on most recent job
  - or specify job number with `%num`

# Process IDs

- Each process is assigned a number (process id, or "pid")
- `ps`
  - display processes
  - use "ax" to show all processes on machine
    - `ps ax`
    - `ps ax | grep firefox`
  - many more options
- `pidof <processname>`
  - `pidof zwgc`

# Killing Processes

- `kill 123`
- `kill -SIGNAL 123`
  - `kill -9 123`
- Must be root to kill other users' processes
- `pkill <process name>`
  - `pkill zwgc`
  - "punt" in the consult locker does the same thing

# For more information

- O'Reilly Technical Books
  - <http://libraries.mit.edu/get/safari>
  - "Learning the bash Shell"
- <http://kb.mit.edu/confluence/display/ist/Working+with+the+Command+Line>

# More Minicourses

- Sep 6: Web Publishing on Athena
- Sep 7: Dotfiles and Shell Customizations
  
- <http://mit.edu/ist/helpdesk/minicourses/>