

Don't Panic

Spring, 2007

MTA-Tech

Dave Pacheco (dap@cs.brown.edu)

Summary

- **Introduction**
- Filesystem
- Using the tools
- File permissions
- Miscellaneous

Introduction

- UTA's spend more time than most undergrads working on departmental systems
- UTA's use more parts of the system than most undergrads
- Consequences of mistakes are larger in scope
 - releasing solutions, grades, exams publicly
 - messing up grades

Summary

- Introduction
- **Filesystem**
- Using the tools
- File permissions
- Miscellaneous

Filesystem

- Filesystem is a tree
 - like Windows “C” drive
 - but there are no other “drives” - just one FS
- *Paths* are a sequence of directories to find a particular file, separated by '/'
 - e.g., /course/cs017/admin/ta-salaries.txt
- One directory is your *working directory* – everything is relative to that (use leading '/' to make paths absolute)

Filesystem: important places

- */home/user/ == ~user/ (just ~ for you)*
 - Files/folders starting with '.' are “hidden”
 - Most of your preferences are stored in “dotfiles” in your home directory
 - .vimrc, .cshrc, .alias, .environment
- */course/csXXX/*
 - different for each course, but usually includes asgn, handin, web, etc.
- “.” (current directory)
- “..” (parent directory)

Filesystem: snapshots

- Sort of like a backup
- Every directory has a hidden “.snapshot” directory
- Each snapshot directory has hourly, nightly, and weekly subdirectories
- Each of these contains the contents of the original folder at that time
 - nightly.0 = last night
 - hourly.1 = 8 hours ago, etc.

Summary

- Introduction
- Filesystem
- **Using the tools**
- File permissions
- Miscellaneous

Using tools

- Usual syntax: *program options arguments*
 - e.g., `grep -i -r "lambda" /course/cs017`
- To find out about syntax and various options, use *man commandname*
 - e.g., `man grep`
 - use this *frequently*

Some useful tools

- `man(1)` – manual pages
- `cd` – change directory
- `ls(1)` – list files in a directory
- `pwd(1)` – print working directory
- `mv(1)` – move files
- `cp(1)` – copy files
- see “cheat sheet”

Shell magic

- *, ? are wildcards which refer to any number of any character or 1 of any character, respectively
 - *e.g.*, *.c == all files ending in “.c”
- Type “&” after a command name to run it in the background (allowing you to run other commands)
 - *e.g.*, xpdf foo.pdf &

Job control

- Press CTRL-C to kill a running process.
- Press CTRL-Z to suspend a running process.
- “jobs” shows you running background processes (jobs) (%X refers to job #X)
- Use “kill” to kill background jobs
- Use “fg” and “bg” to move jobs to foreground and background, respectively
 - “bg” resumes a job you suspended, but in the background.

Manipulating input and output

- Use | (pipe) to connect output of one command to input of another
 - *e.g.*, `find ~ -name "*.o" | grep kern`
(prints files ending in .o whose names contain kern)
- Use > or < to redirect input/output to/from files
 - *e.g.*, `cat filename > newfile` (poor man's "cp")

Summary

- Introduction
- Filesystem
- Using the tools
- **File permissions**
- Miscellaneous

File permissions: intro

- Each file and directory has *permissions*
- 3 classes of people, 3 permissions each
 - owner, owning group, everybody else
 - read, write, execute
 - always described in that order
- “ls -l” shows permissions as:
 - `rw-rw-rwx` (everybody can r/w/x)
 - `rw-rw----` (owner and group can r/w)
 - `rw-----` (owner can r/w)

File permissions: meaning

- Files

- read = can **view** contents of file
- write = can **change** contents of file
- execute = can **run** file as a program

- Directories

- read = can list files in the directory
- write = can create, rename, delete files from the directory
- execute = can follow paths through directory

File permissions: changing

- `chmod(1)` changes file permissions
- `chmod [args] newmode files...`
 - most useful arg is “-R” = recursive
- `newmode` can be numeric or *symbolic*

File permissions: numeric

- Sometimes, permissions represented as 3 octal (base-8) digits
 - 1st digit = user, 2nd digit = group
 - 3rd digit = everyone else
- Each digit is sum of
 - 4 = read
 - 2 = write
 - 1 = execute
- *e.g.*, 775 (rwxrwxr-x)

File permissions: symbolic perms

- symbolic is 1 or more of ugoa, plus 1 of +/-=, plus 1 or more of ugorwx (and a few others)
 - ugoa = user, group, other, all
 - + adds, - removes, = sets

File permissions: examples

- `chmod ug+rw file1 file2 file3`
 - Allow user and group to read and write (doesn't affect execute bit)
- `chmod g=u file1 file2`
 - Make group perms same as user perms
- `chmod o= file1 file2`
 - Set permissions to "" (removes all privileges) for "other people"

File permissions: usage

- Most files in the course directories should be `g=u` (any TA should be able to do what the owner is allowed to do)
 - this is usually read/write
- Programs should have execute bit
- Files not intended for students should have none of `rx` on “other”, and should have group “`csXXXta`”
 - use `chgrp(1)` to change groups

File permissions: defaults

- Files you create are owned by you and belong to the group 'ugrad' by default or, if sticky bit is set on the parent directory, the group of the parent directory
- Course directories should have this bit set
- `man chmod(1)` for more details on the sticky bit

File permissions: defaults (2)

- Permissions on files you create are everything MINUS your 'umask' – (default: 022).
- Use 'umask' command to set umask to 007 in the course directory so new files have no permissions for anyone other than the user and group.
- man umask(1), umask(2) (those are man pages 1 and 2 for umask – use “man 1 umask” or “man 2 umask” to see them)

File permissions: Windows

- Don't ever use them.
- Changing the Windows permissions of a file breaks the Linux ones (which most people use most of the time).
- Always log into a Linux system (or use a Cygwin shell from a Windows system) and use `chmod(1)` to change permissions.

Summary

- Introduction
- Filesystem
- Using the tools
- File permissions
- **Miscellaneous**

Email

- Should check your email daily
- Forward email using
`mailconfig -a you@whatever.com`
- To read mail, use pine from command line, or thunderbird, etc.

Identities

- Identities determine disk quotas
- Can also affect important environment variables, handin scripts, etc.
- Make sure you have csXXXta identity
- Not directly related to groups, which control permissions
- Make sure you have csXXXta group

Remote access

- ssh: allows you to get a shell from a computer outside the department
- Can also access email and shells remotely, using VPN, ssh, or web, etc.
- See CS Department Documentation
<http://www.cs.brown.edu/facilities/system>

Miscellaneous

- Test accounts: csXXX000 (ask HTA or Dorinda (djm))
- /pro/tasupport has useful programs
- Learn LaTeX, vim, advanced commands with Sunlab minicourses

Resources

- Sunlab consultants (sit at 9a)
- mta@cs.brown.edu (Meta-TA tech): email this first with random technical questions (related to TA'ing)
- problem@cs.brown.edu: if you have problems with the departmental systems
- <http://www.cs.brown.edu/facilities/system>
Great documentation on everything
- <http://wiki.cs.brown.edu/>
Internal Wiki (lots of useful info)