

CS-344 - Unix Operating System Fundamentals

Lecture 4
Accessing and Changing Previous Commands
&
Setting File and Directory Permissions

Based on slides created by
Dr. Bangalore for the
Spring 2005 offering of
the course

history (I)

- ❑ Displays a list of previous commands executed by the shell
- ❑ Each command has an associated number
- ❑ To repeat the last command enter !!
- ❑ To execute a command by event number use !*number* (e.g., !19 executes the command associated with event # 19)
- ❑ To execute a command that begins with a specific string use !*string* (e.g., !ca will execute the last command starting with ca)

history (II)

- ❑ To add history event number to command prompt enter `export PS1='[\!] $ '`
- ❑ To select all arguments from previous command use `!*` as argument of new command

```
cat quizscores homework
wc !*
```

- ❑ To select the last argument of previous command use `!$` as argument of new command
- ❑ To add an argument to a previous command use `!! NewArgument` or `!string NewArgument`

```
!! -l
!cat hw5
```

6/23/2005

4

File Permissions

- ❑ Typical UNIX user performs the following operations on files:
 - Read files (using `more`, `cat`, etc.)
 - Write files (using `>`, `>>`, `cat`, `vi`, etc.)
 - Execute commands in a file (shell scripts, executables, etc.)
- ❑ Correspondingly each file has three permissions read, write, and execute (`rwX`)
- ❑ On UNIX systems there are three classes of users: the *owner*, other members of owner's *group*, and all *other* users
- ❑ The owner can modify permissions for each of these three classes of users
- ❑ To examine file permissions use `ls -l`

```
$ ls -l myfile
-rw-r--r-- 1 puri staff 2093 Feb 28 10:52 myfile
$ ls -ld cs344
drwxr-xr-x 3 puri staff 4096 Feb 23 12:02 cs344
```

6/23/2005

5

Determine user and group

- ❑ To determine login name of user type `echo $USER` or `who am i`
- ❑ To determine what groups you belong type `groups` (first group is your default group)
- ❑ To change to a new group type `newgrp groupname`, any new files created now will have this group name
- ❑ To determine your user id (UID) and group id (GID) type `id`
- ❑ To change group use the command `chgrp` and to change owner use the command `chown`

6/23/2005

6

Changing File Permissions (I)

- To change current file permissions use “*chmod*” (change mode) command
- To add specific permission use *chmod +*
 - To add write permission to all users use:
chmod a+w filename
 - To add read permission to only to users in your group use: *chmod g+r filename*
- To remove specific permission use *chmod -*
 - To remove read permission for all users use:
chmod a-r filename
 - To remove read, write, and execute permission for the group and others use: *chmod go-rwx filename*
- You can also combine add and remove permissions (e.g., *chmod u+x,g+r,o-rwx filename*)

6/23/2005

7

Changing File Permissions (II)

- Summary:

Types of Users		Permissions		Actions	
u	user	r	read	=	assign
g	group	w	write	-	remove
o	other	x	execute	+	add
a	all	-	none		

6/23/2005

8

Using numerical permissions

- Instead of using u,g,o for user, group, and others we can also specify file permissions using numbers:
 - rx* = 11 = 7
 - rw* = 110 = 6
 - r-x* = 101 = 5
 - r--* = 100 = 4
 - wx* = 011 = 3
 - w-* = 010 = 2
 - x* = 001 = 1
 - = 000 = 0
- *chmod go+rx filename = chmod 755 filename* (assuming current user permission is *rx* = 7, if it is *rw-* = 6, then use *chmod 655 filename*)

6/23/2005

9

Directory Permissions

- ❑ To list contents of a directory with ls command we need read permissions
- ❑ To add/remove files in a directory we need write and execute permissions
- ❑ To change to a directory or go through the directory we need execute permissions
- ❑ To list files with ls -l we need read and execute permissions for the directory, since information about permissions, owner, group, etc. are in the directory entry

6/23/2005

10

Set/Get Default Directory Permissions

- ❑ When new file/directory is created the shell uses default permissions determined by umask value
- ❑ To obtain you default umask value, at command prompt enter: umask
- ❑ To change current umask value, enter umask <new-mask-value>
- ❑ Based on the umask value appropriate permissions are unmasked (allowed)
- ❑ Changing umask value has no effect on existing files, only new files will be effected

Umask	New Directory Permissions	
000	rwxrwxrwx	777
022	rwxr-xr-x	755
027	rwxr-x---	750
017	rwxrw---	760

6/23/2005

11

Get/Set File Permissions

- ❑ Set umask to 000, create a new file, list the file using ls -l, this will indicate the default file permission (typically rw-rw-rw- = 666)
- ❑ Execute permission are never granted when files are created hence setting the mask on execute bit has no effect
- ❑ Set umask to 022, create a new file, list file, the new file permissions will be rw-r--r-- = 644
- ❑ Set umask to 023, create a new file, list file, the new file permission will be still rw-r--r--
- ❑ To retain file permissions during file copy use cp -p option

6/23/2005

12

File Permissions

```

$ ls -ld .
drwx--x--x 17 puri faculty 8192 Feb 20 17:24 .
$ ls -l .bashrc
-rwx----- 1 puri staff 1196 Feb 13 19:31 .bash_rc
$ ls -ld /tmp/
drwxrwxrwt 17 root sys 3218 Feb 20 18:35 /tmp/
$ umask
022
$ touch myfile
$ ls -l myfile
-rw-r--r-- 1 puri staff 0 Feb 20 18:38 myfile
$ chmod +x myfile
$ ls -l myfile
-rwxr-xr-x 1 puri staff 0 Feb 20 18:38 myfile
$ umask 027
$ touch newfile
$ ls -l newfile
-rw-r----- 1 puri staff 0 Feb 20 18:39 newfile
$
    
```

6/23/2005

13

Executing Shell Scripts

```

$ cat myscript
echo "Welcome" $USER
echo "Today's date is: " `date | cut -d` ` -f2-3`
echo "You are logged in to: " `hostname`
echo "There are " `who | wc -l` "user(s) currently logged in"
echo "Your PATH is set to the following directories:"
echo $PATH
$ ls -l myscript
-rwxr-xr-x 1 puri staff 239 Feb 20 18:23 myscript
$ myscript
bash: myscript: command not found
Shell did not find the script in it's PATH
$ ./myscript
You can also specify the full path
Welcome puri
Today's date is: Feb 20
You are logged in to: hestia
There are 1 user(s) currently logged in
Your PATH is set to the following directories:
/usr/bin:/usr/sbin:/sbin:/h/local/bin/mz/mb/puri/bin
$ export PATH=$PATH:
You will see the above output this time
$ myscript
    
```

6/23/2005

14
