

Lesson Module Status

- Slides - draft
 - Flash cards – done
 - properties – done
 - page numbers - done
 - 1st minute quiz – done
 - Web Calendar summary – done
 - Web book pages - none
 - Commands – done
 - Lab tested – done
-
- Materials uploaded – done
 - CCC Confer wall paper / quiz - done
 - Check that headset is charged – done
 - Backup headset charged - done
 - Backup slides, CCC info, handouts on flash drive - done



Instructor: **Rich Simms**
Dial-in: **888-450-4821**
Passcode: **761867**



Emanuel



Tanner



Merrick



Quinton



Chris



Bobby



Craig



Jeff



Yu-Chen



Terry



Tommy



Eric



Dan M



Geoffrey



Marisol



David



Josh



Gabriel



Jesse



Tajvia



Daniel W



Jason

Email me (risimms@cabrillo.edu) a relatively current photo of your face for 3 points extra credit

First Minute Quiz

Please close your books, notes, lesson materials, forum and answer these questions **in the order** shown:

- What option for the rm command provides confirmation when deleting files?
- What command is used to rename a file?
- If two files are hard linked do they have the same or different inode numbers?

email answers to: risimms@cabrillo.edu



- [] Has the phone bridge been added?
- [] Is recording on?
- [] Does the phone bridge have the mike?
- [] Share slides, putty (simmsmar, simmsben, roddyduk), and Chrome
- [] Disable spelling on PowerPoint

File Permissions

Objectives	Agenda
• Be able to reassign user and group file ownerships • Identify permissions for ordinary and directory files • Use chmod to set and change file permissions • Define the default permissions for new files	• Quiz • Review test results • Question on previous material • File permissions • Wrap up

Questions

Previous material and assignment

1. Questions on last Lab assignment?
2. Questions on last lesson?
3. Questions on Test #1?

Test 1 – Most Missed Questions

Question# - count of students with missing or incorrect answers

QX4	16	shell parsing
QX2	16	/etc/passwd fields
QX3	14	who command arguments
Q12	8	locating a command file
Q6	6	file types and inodes
Q8	4	path used by shell
Q14	5	shell parsing and filename expansion
QX1	4	using man page
Q15	3	bc and mail commands
Q11	3	PS1 Prompt variable
Q7	2	/etc/password absolute path
Q2	2	cat command
Q3	1	mbox archives
Q10	1	file owner
Q4	1	hostname command
Q9	1	navigation and wc command
Q1	1	xxd (hex dump) command
Q5	0	PATH variable
Q13	0	navigating with cd and ls commands

Test 1 – Most Missed Questions

[QX4] Parse the following command:

cp -p /boot/grub/stage[12] .

*Use the echo command
on Opus to expand the
arguments*

Answer:

Command: **cp**

Options: **-p**

Arguments: **/boot/grub/stage1 /boot/grub/stage2 .**

```
/home/cis90ol/simmsben $ echo /boot/grub/stage[12] .  
/boot/grub/stage1 /boot/grub/stage2 .
```

Note there is one option and three arguments that will be passed to the cp command to process when the cp command is run

Test 1 – Most Missed Questions

[QX2] Which shell program will be started for millehom (Homer Miller) when he logs into Opus?

cat out /etc/passwd and look for the millehom account

```
/home/cis90ol/simmsben $ cat /etc/passwd  
root:x:0:0:root:/root:/bin/bash  
bin:x:1:1:bin:/bin:/sbin/nologin  
daemon:x:2:2:daemon:/sbin:/sbin/nologin
```

< snipped >

```
millehom:x:1116:190:Homer Miller:/home/cis90ol/millehom:/bin/ksh
```

< snipped >

The last field (following the home directory) is the shell to run when that user logs in

Answer: /bin/ksh (the Korn shell)

Test 1 – Most Missed Questions

[QX3] What complete command could display output such as the following, showing idle time and column headers for users currently logged in?

NAME	LINE	TIME	IDLE	PID	COMMENT
rsimms	pts/1	2011-03-15 08:23	.	24017	(dsl-10-78-126-19.dhcp.cruzter.com)
rsimms	pts/2	2011-03-15 07:44	03:41	23499	(dsl-10-78-126-19.dhcp.cruzter.com)
simmsben	pts/3	2011-03-15 08:33	00:04	24083	(dsl-10-78-126-19.dhcp.cruzter.com)
guest130	pts/4	2011-03-15 10:30	.	25354	(adsl-20-201-78-69.dsl.abclocal.net)
delfimik	pts/5	2011-03-15 11:37	.	26996	(adsl-30-27-59-11.dsl.abcglocal.net)
silvetyl	pts/6	2011-03-15 11:38	00:05	27194	(c-40-123-46-9.hsd1.ca.comster.net)
lopezjes	pts/7	2011-03-15 11:21	00:02	25937	(50-75-89-34-sfba.hfc.comster.net)
lopezjes	pts/8	2011-03-15 11:28	00:01	26119	(50-75-89-34-sfba.hfc.comster.net)
vasqucar	pts/9	2011-03-15 10:00	01:30	24903	(50-102-84-123.dhcp.ca.chart.com)

Answer: **who -Hu** or **who -Hi**

The H option is documented with the man command. An option to show idle time is more obscure and not documented on the man page on Opus! You would have to do some Googling or refer back to the example in Lesson 1 slides to answer this question.

Test 1 – Most Missed Questions

[Q12] In what directory does the zforce command reside?

Use the type command to answer this question. Type shows the location on the path where the command is first located.

```
/home/cis9001/simmsben $ type zforce  
zforce is /usr/bin/zforce
```


directory *file*

Answer: **/usr/bin**

Test 1 – Most Missed Questions

[Q6] What file type is the file in the /etc/httpd/conf/ directory with an inode number = 1278942?

Use the ls command with the i option to list the files in that directory with their inode numbers

```
/home/cis90ol/simmsben $ ls -i /etc/httpd/conf  
1278942 httpd.conf 1280028 magic
```

Do a long listing of the file to determine the file type

```
/home/cis90ol/simmsben $ ls -l /etc/httpd/conf/httpd.conf  
-rw-r--r-- 1 root root 33787 Jan 27 16:42 /etc/httpd/conf/httpd.conf
```

The – in column 1 indicates a regular file

Or use the file command to show the file type

```
/home/cis90ol/simmsben $ file /etc/httpd/conf/httpd.conf  
/etc/httpd/conf/httpd.conf: ASCII English text
```

Answer: **“regular file” or “ASCII English text” or both**

Test 1 – Most Missed Questions

[Q8] Benji issues the following command to show there is more than one banner command on Opus:

```
/home/cis90ol/simmsben $ ls -l bin/banner /usr/local/bin/banner
-rwxr-xr-x 1 simmsben cis90ol 6160 Aug 28 2003 bin/banner
-rwxr-xr-x 1 root      root    11235 Jan 9 2010 /usr/local/bin/banner
```

```
/home/cis90ol/simmsben $ banner hi benji
```

```
#      #  #####
#      #      #
#      #      #
#####      #
#      #      #
#      #      #
#      #  #####
```

```
#####  ##### #      #      #  #####
#      # #      ##      #      #
#      # #      # #      #      #
#####  #####      # #      #      #
#      # #      #      # # #      #      #
#      # #      #      ## #      #      #
#####  ##### #      #  #####  #####
```

```
/home/cis90ol/simmsben $
```

Use the type command to show the location of the banner command that would be run

```
/home/cis90ol/simmsben $ type banner
banner is /usr/local/bin/banner
```

Answer: /usr/local/bin

When Benji types the **banner Hi Benji** command from his home directory, which banner command did he run? The one in **/usr/local/bin** or the one in his own **bin** directory?

Test 1 – Most Missed Questions

[Q8] Continued ...

There is a banner command in both the /usr/local/bin directory and your local bin directory:

```
/home/cis900l/simmsben $ ls -l bin/banner /usr/local/bin/banner
-rwxr-xr-x 1 simmsben cis900l  6160 Aug 28  2003 bin/banner
-rwxr-xr-x 1 root      root    11235 Jan  9  2010 /usr/local/bin/banner
```

The shell will search for a command by looking, in order, in each directory on the path. The banner command in /usr/local/bin is found before the one in /home/cis900l/simmsben/bin

```
/home/cis900l/simmsben $ echo $PATH
/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:/home/cis900l/simmsben/
../bin:/home/cis900l/simmsben/bin:.
```

An easier way to see what command the shell will load is to use the type command:

```
/home/cis900l/simmsben $ type banner
banner is /usr/local/bin/banner
```

Answer: /usr/local/bin

Test 1 – Most Missed Questions

[Q14] When the command **ls /usr/bin/z[if]????** is run from your home directory, what arguments does the shell provide to the **ls** command to process?

Use the echo command on Opus to expand the arguments

Parse the command line:

Command: **ls**

Options: na

Arguments: **/usr/bin/zfgrep /usr/bin/zforce**

```
/home/cis900l/simmsben $ echo /usr/bin/z[if]????  
/usr/bin/zfgrep /usr/bin/zforce
```

Answer: **/usr/bin/zfgrep /usr/bin/zforce**

The shell first parses the command line and identifies all options and arguments. Next the shell locates the command to run. The command, when it is loaded and run, is provided with all the parsed options and arguments.

Test 1 – Most Missed Questions

[QX1] What command (shown as ?????? below) would produce the output below?

```
/home/cis90ol/simmsben $ ??????  
xxd V1.10 27oct98 by Juergen Weigert  
/home/cis90ol/simmsben
```

xxd provides the clue to which command it is. Use the man page to see that the v option shows the version. Try it and you are done with this question!

Answer: **xxd -v**

Test 1 – Most Missed Questions

[Q15] Calculate: $2 + (23^2 * 4 - 99) - \sqrt{64}$
and email the answer to **rsimms** using the **mail** command.

First calculate the answer using the bc (bianry calculator) command

```
/home/cis90ol/simmsben $ bc
```

```
bc 1.06
```

```
Copyright 1991-1994, 1997, 1998, 2000 Free Software Foundation, Inc.
```

```
This is free software with ABSOLUTELY NO WARRANTY.
```

```
For details type `warranty'.
```

```
2 + (23^2 * 4 - 99) - sqrt(64)
```

```
2011
```

```
quit
```

```
/home/cis90ol/simmsben $ bc
```

Then mail the answer to rsimms

```
/home/cis90ol/simmsben $ mail rsimms
```

```
Subject: Q15
```

```
2011
```

```
.
```

```
Cc: simmsben
```

```
/home/cis90ol/simmsben $
```

Test 1 – Most Missed Questions

[Q11] What command will show how your primary prompt is configured?

Display the value of the PS1 variable

```
/home/cis90ol/simmsben $ echo $PS1  
$PWD $
```

Answer: **echo \$PS1**

Test 1 – Most Missed Questions

[Q7] What is the actual absolute pathname to the /etc/passwd file?

Answer: **/etc/passwd**

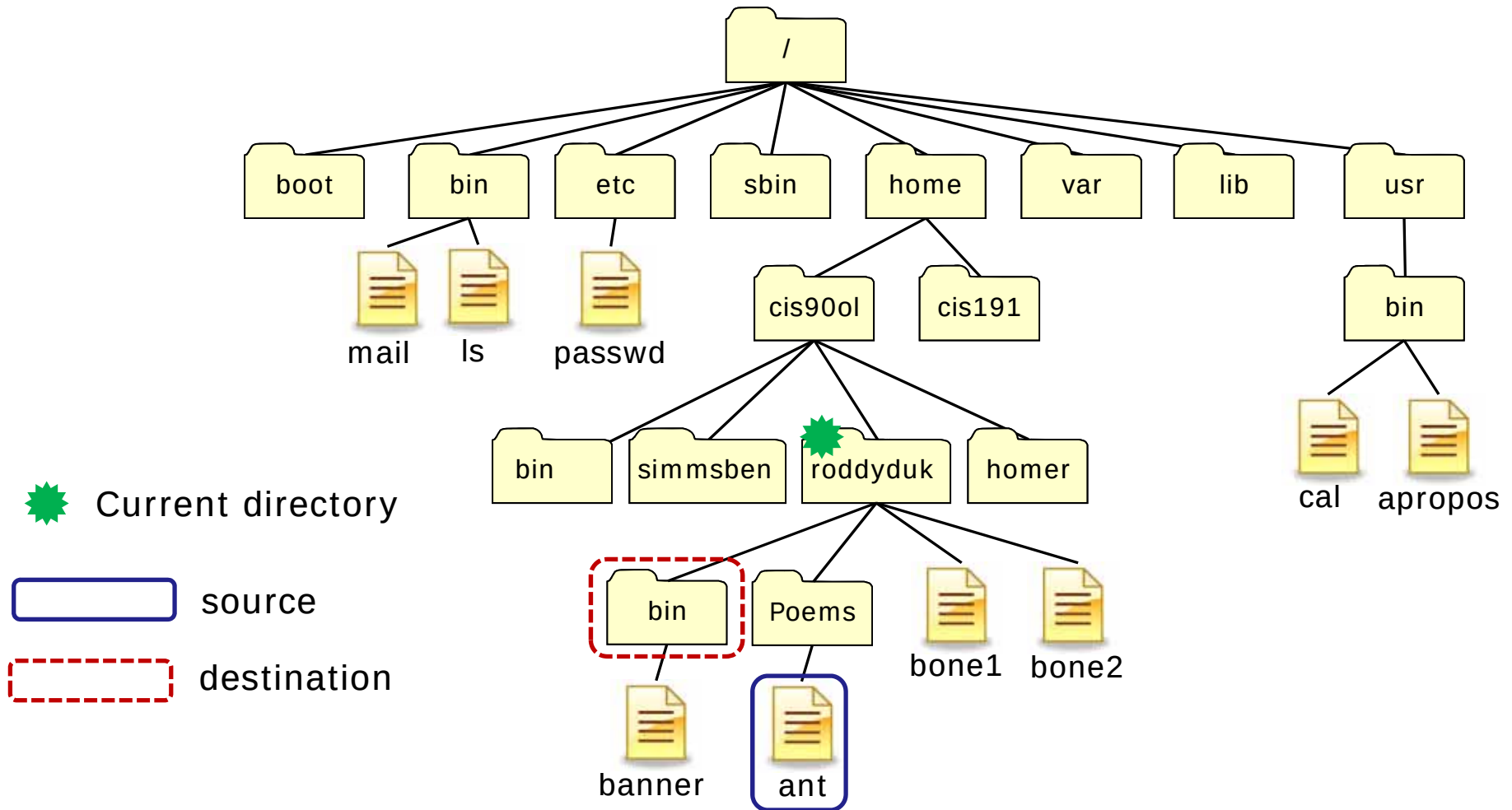
What is the color of George Washington's white horse?

Who is buried in Grant's tomb?

This question will be on a future test!

Pathname practice

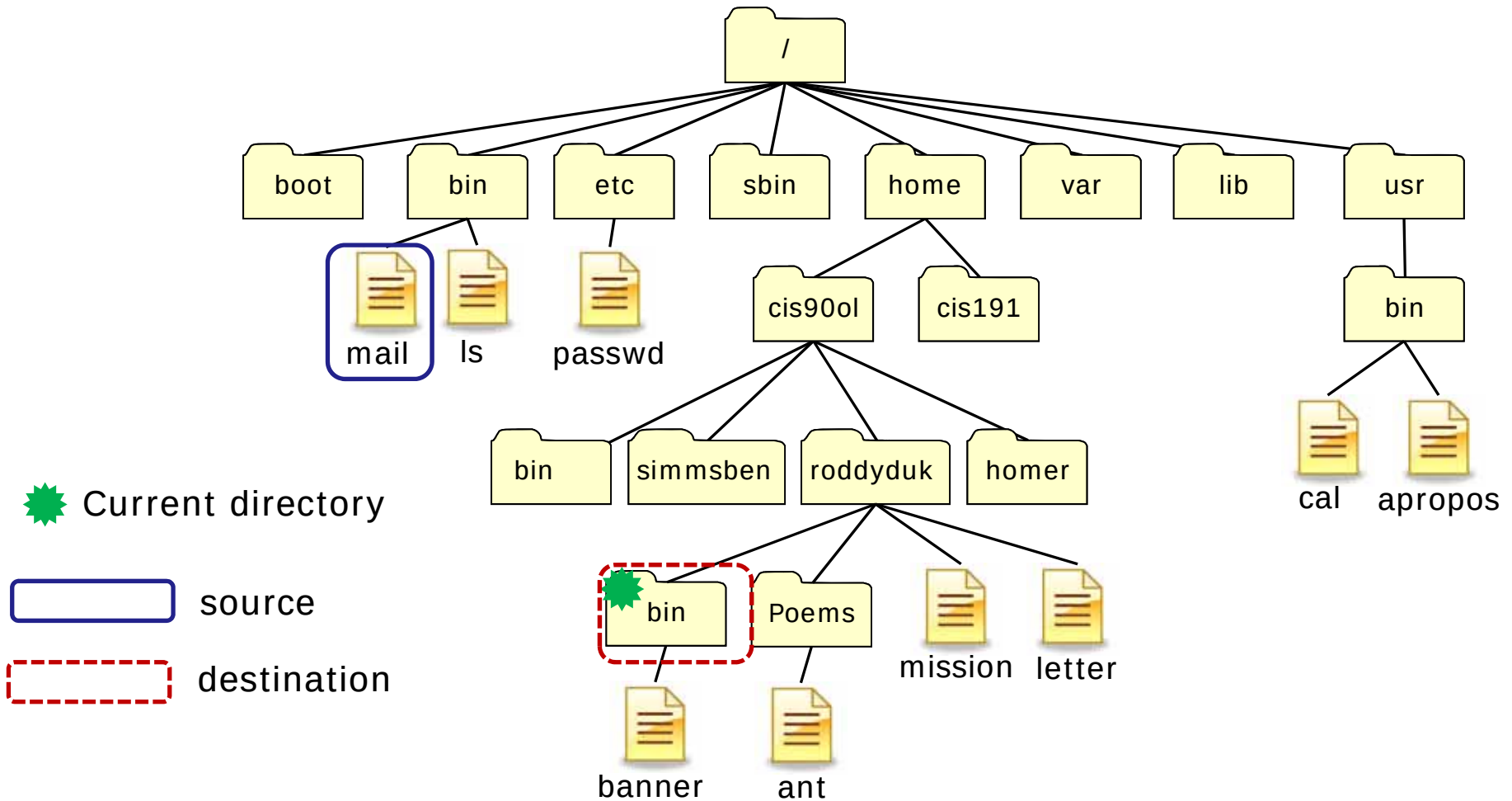
Pathname Practice



What command copies the ant file to the bin directory as shown above?

`cp Poems/ant bin/`

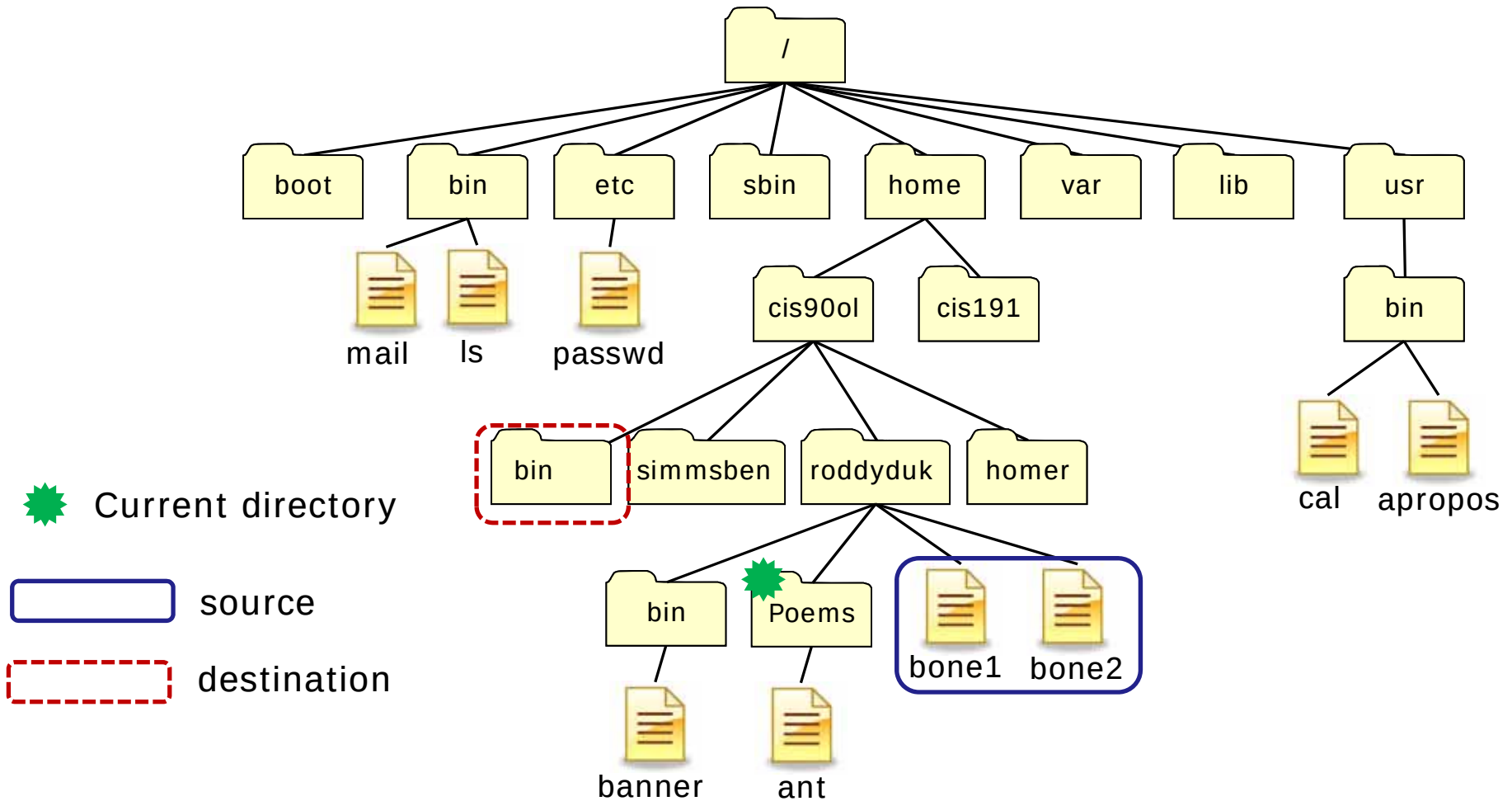
Pathname Practice



What command copies the mail program in /bin to your current working directory?

`cp /bin/mail .`

Pathname Practice



What command moves the bone1 and bone2 files to the bin directory shown?

`mv ../bone? ../../bin/`

Housekeeping

Housekeeping

- Tutoring is still available
 - Contact Jack Hsieh at hsiehrr@yahoo.com
- Perkins (VTEA) Web Advisor Survey
 - The information collected in the survey will enable the college to qualify for grant funds to support instruction.
 - **All of the information is confidential.**
 - Each student should fill out **one survey per semester**. If students have filled out this form in another class this semester, **it is not necessary to fill out the survey more than once per semester** unless some information has changed.

Perkins (VTEA) Web Advisor Survey Instructions

If you already filled this out in another class you don't need to do it again. This is the online survey for online classes.

- Log on to “www.cabrillo.edu” and go to the Cabrillo College Home Page
 - Select “WEBADVISOR” (from the listing near the top of the page below the Cabrillo College Logo)
 - Select the “LOG IN” tab
 - Fill-in the “User ID” and “Password”
 - Click on “SUBMIT”
- Select “STUDENTS: Click Here” (navy blue bar)
 - Under “Academic Profile” Click on “Student Update Form”
 - Use drop down list under “Select the earliest term for which you are registered” and click on the current term (Spring 2011).
 - Select “SUBMIT”
- Scroll down to the “Career Technical Information”
 - **Answer questions** by clicking on the circle to the left of your “Yes” or “No” answers
 - You can get details about a question by clicking on blue underlined phrase
 - After answering all questions Select “SUBMIT”
 - Then “LOG OUT”

Career Technical Information
Your answers to these questions will help qualify Cabrillo College for Perkins/VTEA grant funds.

Are you currently receiving benefits from:

☐ YES ☐ NO TAFICALWORKS

☐ YES ☐ NO SSI (Supplemental Security Income)

☐ YES ☐ NO GA (General Assistance)

☐ YES ☐ NO Does your income qualify you for a fee waiver?

☐ YES ☐ NO Are you a single parent with custody of one or more minor children?

☐ YES ☐ NO Are you a displaced homemaker attending Cabrillo to develop job skills?

☐ YES ☐ NO Have you moved in the preceding 36 months to obtain, or to accompany parents or spouses to obtain, temporary or seasonal employment in agriculture, dairy, or fishing?

Thank you for taking a few minutes to help Cabrillo receive funding to support student services for CTE programs at Cabrillo College.

one more
mv



Home directory cleanup

- Move your graded work to /class/labs

`mv test01.graded class/exams/`

- Check with ls command

Permissions



File Permissions



Permissions are use to control access to files

Benji has read permission to /etc/passwd

```
/home/cis90ol/simmsben $ head -3 /etc/passwd  
root:x:0:0:root:/root:/bin/bash  
bin:x:1:1:bin:/bin:/sbin/nologin  
daemon:x:2:2:daemon:/sbin:/sbin/nologin
```

But he does not have read permission to /etc/shadow!

```
/home/cis90ol/simmsben $ head -3 /etc/shadow  
head: cannot open `/etc/shadow' for reading: Permission denied  
/home/cis90ol/simmsben $
```



File Permissions



Permissions are use to control access to files

Benji had write permission to his own letter file

This is how you append a line to a file. We will learn more about >> later!

```
/home/cis90ol/simmsben $ echo "Benji was here" >> letter
/home/cis90ol/simmsben $ tail -4 letter
Mother, Father, kindly disregard this letter.
```

Alan Sherman

```
Benji was here
```

But he does not have write permission to Duke's letter file!

```
/home/cis90ol/simmsben $ echo "Benji was here" >> ../roddyduk/letter
-bash: ../roddyduk/letter: Permission denied
/home/cis90ol/simmsben $
```



File Permissions



Permissions are use to control access to files

Benji has execute permission on his tryme script and he runs it

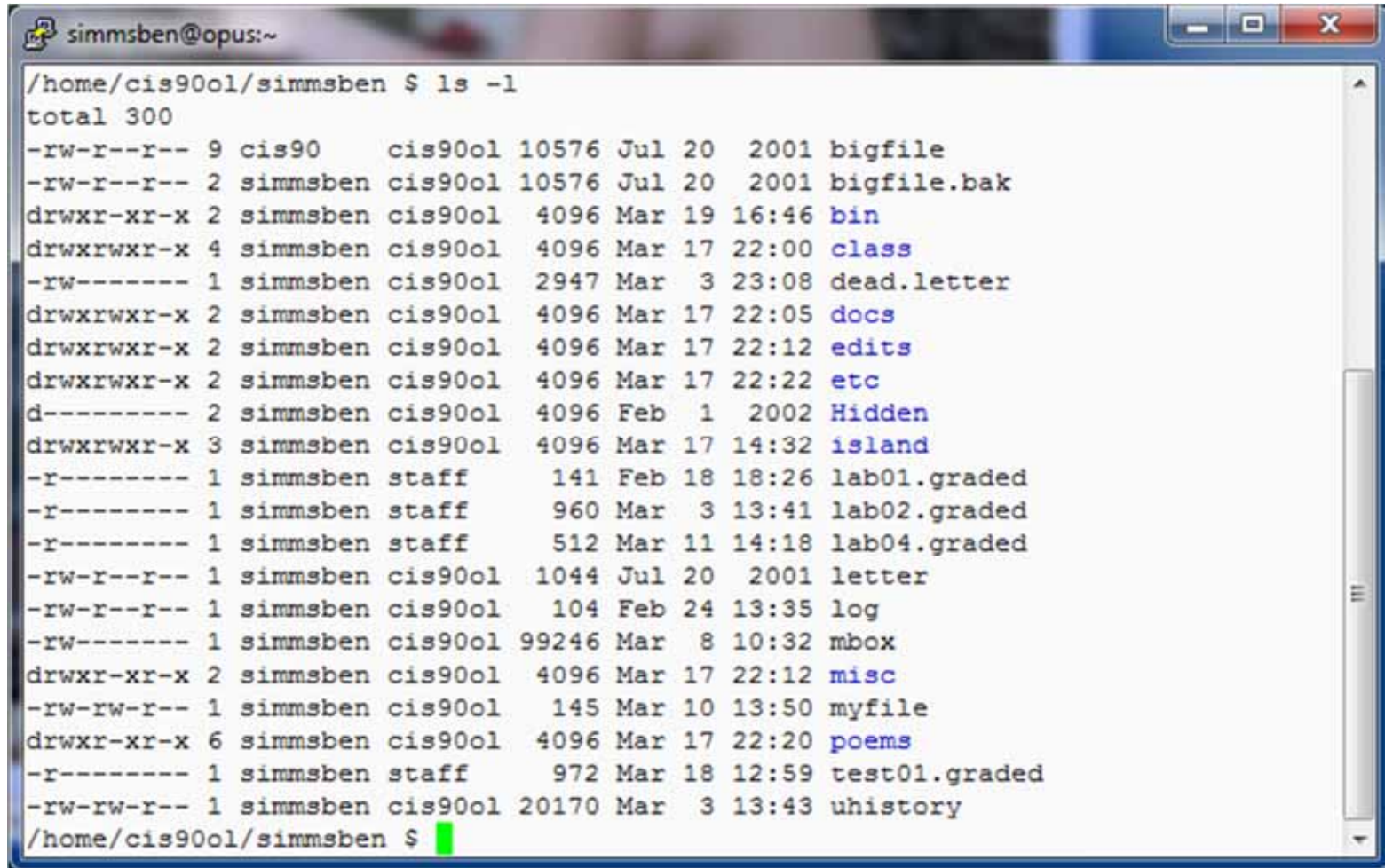
```
/home/cis90ol/simmsben $ tryme  
My name is "tryme"  
I am pleased to make your acquaintance, Benji Simms  
/tmp  
/home/cis90ol/simmsben $
```

But he does not have execute permission on the check7 script so he can't run it!

```
/home/cis90ol/simmsben $ check7  
-bash: /home/cis90ol/simmsben/../../bin/check7: Permission denied
```

File Permissions

Interpreting the permission codes on long listings



```
simmsben@opus:~  
/home/cis90ol/simmsben $ ls -l  
total 300  
-rw-r--r-- 9 cis90    cis90ol 10576 Jul 20  2001 bigfile  
-rw-r--r-- 2 simmsben cis90ol 10576 Jul 20  2001 bigfile.bak  
drwxr-xr-x 2 simmsben cis90ol  4096 Mar 19 16:46 bin  
drwxrwxr-x 4 simmsben cis90ol  4096 Mar 17 22:00 class  
-rw----- 1 simmsben cis90ol  2947 Mar  3 23:08 dead.letter  
drwxrwxr-x 2 simmsben cis90ol  4096 Mar 17 22:05 docs  
drwxrwxr-x 2 simmsben cis90ol  4096 Mar 17 22:12 edits  
drwxrwxr-x 2 simmsben cis90ol  4096 Mar 17 22:22 etc  
d----- 2 simmsben cis90ol  4096 Feb  1  2002 Hidden  
drwxrwxr-x 3 simmsben cis90ol  4096 Mar 17 14:32 island  
-r----- 1 simmsben staff    141 Feb 18 18:26 lab01.graded  
-r----- 1 simmsben staff    960 Mar  3 13:41 lab02.graded  
-r----- 1 simmsben staff    512 Mar 11 14:18 lab04.graded  
-rw-r--r-- 1 simmsben cis90ol  1044 Jul 20  2001 letter  
-rw-r--r-- 1 simmsben cis90ol   104 Feb 24 13:35 log  
-rw----- 1 simmsben cis90ol 99246 Mar  8 10:32 mbox  
drwxr-xr-x 2 simmsben cis90ol  4096 Mar 17 22:12 misc  
-rw-rw-r-- 1 simmsben cis90ol   145 Mar 10 13:50 myfile  
drwxr-xr-x 6 simmsben cis90ol  4096 Mar 17 22:20 poems  
-r----- 1 simmsben staff    972 Mar 18 12:59 test01.graded  
-rw-rw-r-- 1 simmsben cis90ol 20170 Mar  3 13:43 uhistory  
/home/cis90ol/simmsben $
```

Use a long listings to view file permissions

File Permissions

Interpreting the permission codes on long listings

```
simmsben@opus:~
/home/cis90ol/simmsben $ ls -l
total 300
-rw-r--r-- 9 cis90 cis90ol 10576 Jul 20 2001 bigfile
-rw-r--r-- 2 simmsben cis90ol 10576 Jul 20 2001 bigfile.bak
drwxr-xr-x 2 simmsben cis90ol 4096 Mar 19 16:46 bin
drwxrwxr-x 4 simmsben cis90ol 4096 Mar 17 22:00 class
-rw----- 1 simmsben cis90ol 2947 Mar 3 23:08 dead.letter
drwxrwxr-x 2 simmsben cis90ol 4096 Mar 17 22:05 docs
drwxrwxr-x 2 simmsben cis90ol 4096 Mar 17 22:12 edits
drwxrwxr-x 2 simmsben cis90ol 4096 Mar 17 22:22 etc
d----- 2 simmsben cis90ol 4096 Feb 1 2002 Hidden
drwxrwxr-x 3 simmsben cis90ol 4096 Mar 17 14:32 island
-r----- 1 simmsben staff 141 Feb 18 18:26 lab01.graded
-r----- 1 simmsben staff 960 Mar 3 13:41 lab02.graded
-r----- 1 simmsben staff 512 Mar 11 14:18 lab04.graded
-rw-r--r-- 1 simmsben cis90ol 1044 Jul 20 2001 letter
-rw-r--r-- 1 simmsben cis90ol 104 Feb 24 13:35 log
-rw----- 1 simmsben cis90ol 99246 Mar 8 10:32 mbox
drwxr-xr-x 2 simmsben cis90ol 4096 Mar 17 22:12 misc
-rw-rw-r-- 1 simmsben cis90ol 145 Mar 10 13:50 myfile
drwxr-xr-x 6 simmsben cis90ol 4096 Mar 17 22:20 poems
-r----- 1 simmsben staff 972 Mar 18 12:59 test01.graded
-rw-rw-r-- 1 simmsben cis90ol 20170 Mar 3 13:43 uhistory
/home/cis90ol/simmsben $
```

This column shows the permissions on each file

File Permissions

Interpreting the permission codes on long listings

```
simmsben@opus:~
/home/cis90ol/simmsben $ ls -l
total 300
-rw-r--r--  9 cis90    cis90ol 10576 Jul 20  2001 bigfile
-rw-r--r--  2 simmsben cis90ol 10576 Jul 20  2001 bigfile.bak
drwxr-xr-x  2 simmsben cis90ol  4096 Mar 19 16:46 bin
drwxrwxr-x  4 simmsben cis90ol  4096 Mar 17 22:00 class
-rw-r----- 1 simmsben cis90ol  2947 Mar  3 23:08 dead.letter
drwxrwxr-x  2 simmsben cis90ol  4096 Mar 17 22:05 docs
drwxrwxr-x  2 simmsben cis90ol  4096 Mar 17 22:12 edits
drwxrwxr-x  2 simmsben cis90ol  4096 Mar 17 22:22 etc
dr-----  2 simmsben cis90ol  4096 Feb  1  2002 Hidden
drwxrwxr-x  3 simmsben cis90ol  4096 Mar 17 14:32 island
-r-----  1 simmsben staff    141 Feb 18 18:26 lab01.graded
-r-----  1 simmsben staff    960 Mar  3 13:41 lab02.graded
-r-----  1 simmsben staff    512 Mar 11 14:18 lab04.graded
-rw-r--r--  1 simmsben cis90ol  1044 Jul 20  2001 letter
-rw-r--r--  1 simmsben cis90ol   104 Feb 24 13:35 log
-rw-r----- 1 simmsben cis90ol 99246 Mar  8 10:32 mbox
drwxr-xr-x  2 simmsben cis90ol  4096 Mar 17 22:12 misc
-rw-rw-r--  1 simmsben cis90ol   145 Mar 10 13:50 myfile
drwxr-xr-x  6 simmsben cis90ol  4096 Mar 17 22:20 poems
-r-----  1 simmsben staff    972 Mar 18 12:59 test01.graded
-rw-rw-r--  1 simmsben cis90ol 20170 Mar  3 13:43 uhistory
/home/cis90ol/simmsben $
```

This column shows the owner of each file

File Permissions

Interpreting the permission codes on long listings

```
simmsben@opus:~
/home/cis90ol/simmsben $ ls -l
total 300
-rw-r--r-- 1 simmsben cis90ol 10576 Jul 20 2001 bigfile
-rw-r--r-- 1 simmsben cis90ol 10576 Jul 20 2001 bigfile.bak
drwxr-xr-x 2 simmsben cis90ol 4096 Mar 19 16:46 bin
drwxr-xr-x 4 simmsben cis90ol 4096 Mar 17 22:00 class
-rw-r--r-- 1 simmsben cis90ol 2947 Mar 3 23:08 dead.letter
drwxr-xr-x 2 simmsben cis90ol 4096 Mar 17 22:05 docs
drwxr-xr-x 2 simmsben cis90ol 4096 Mar 17 22:12 edits
drwxr-xr-x 2 simmsben cis90ol 4096 Mar 17 22:22 etc
drwxr-xr-x 2 simmsben cis90ol 4096 Feb 1 2002 Hidden
drwxr-xr-x 3 simmsben cis90ol 4096 Mar 17 14:32 island
-rw-r--r-- 1 simmsben staff 141 Feb 18 18:26 lab01.graded
-rw-r--r-- 1 simmsben staff 960 Mar 3 13:41 lab02.graded
-rw-r--r-- 1 simmsben staff 512 Mar 11 14:18 lab04.graded
-rw-r--r-- 1 simmsben cis90ol 1044 Jul 20 2001 letter
-rw-r--r-- 1 simmsben cis90ol 104 Feb 24 13:35 log
-rw-r--r-- 1 simmsben cis90ol 99246 Mar 8 10:32 mbox
drwxr-xr-x 2 simmsben cis90ol 4096 Mar 17 22:12 misc
-rw-r--r-- 1 simmsben cis90ol 145 Mar 10 13:50 myfile
drwxr-xr-x 6 simmsben cis90ol 4096 Mar 17 22:20 poems
-rw-r--r-- 1 simmsben staff 972 Mar 18 12:59 test01.graded
-rw-r--r-- 1 simmsben cis90ol 20170 Mar 3 13:43 uhistory
/home/cis90ol/simmsben $
```

This column shows the group each file belong to

File Permissions

Interpreting the permission codes on long listings

```
simmsben@opus:~
/home/cis90ol/simmsben $ ls -l
total 300
-rw-r--r-- 9 cis90 cis90ol 10576 Jul 20 2001 bigfile
-rw-r--r-- 2 simmsben cis90ol 10576 Jul 20 2001 bigfile.bak
drwxr-xr-x 2 simmsben
drwxrwxr-x 4 simmsben
-rw----- 1 simmsben
drwxrwxr-x 2 simmsben
drwxrwxr-x 2 simmsben
drwxrwxr-x 2 simmsben
d----- 2 simmsben
drwxrwxr-x 3 simmsben
-r----- 1 simmsben
-r----- 1 simmsben
-r----- 1 simmsben
-rw-r--r-- 1 simmsben
-rw-r--r-- 1 simmsben
-rw----- 1 simmsben
drwxr-xr-x 2 simmsben
-rw-rw-r-- 1 simmsben
drwxr-xr-x 6 simmsben cis90ol 4096 Mar 17 22:20 poems
-r----- 1 simmsben staff 972 Mar 18 12:59 test01.graded
-rw-rw-r-- 1 simmsben cis90ol 20170 Mar 3 13:43 uhistory
/home/cis90ol/simmsben $
```

owner group others

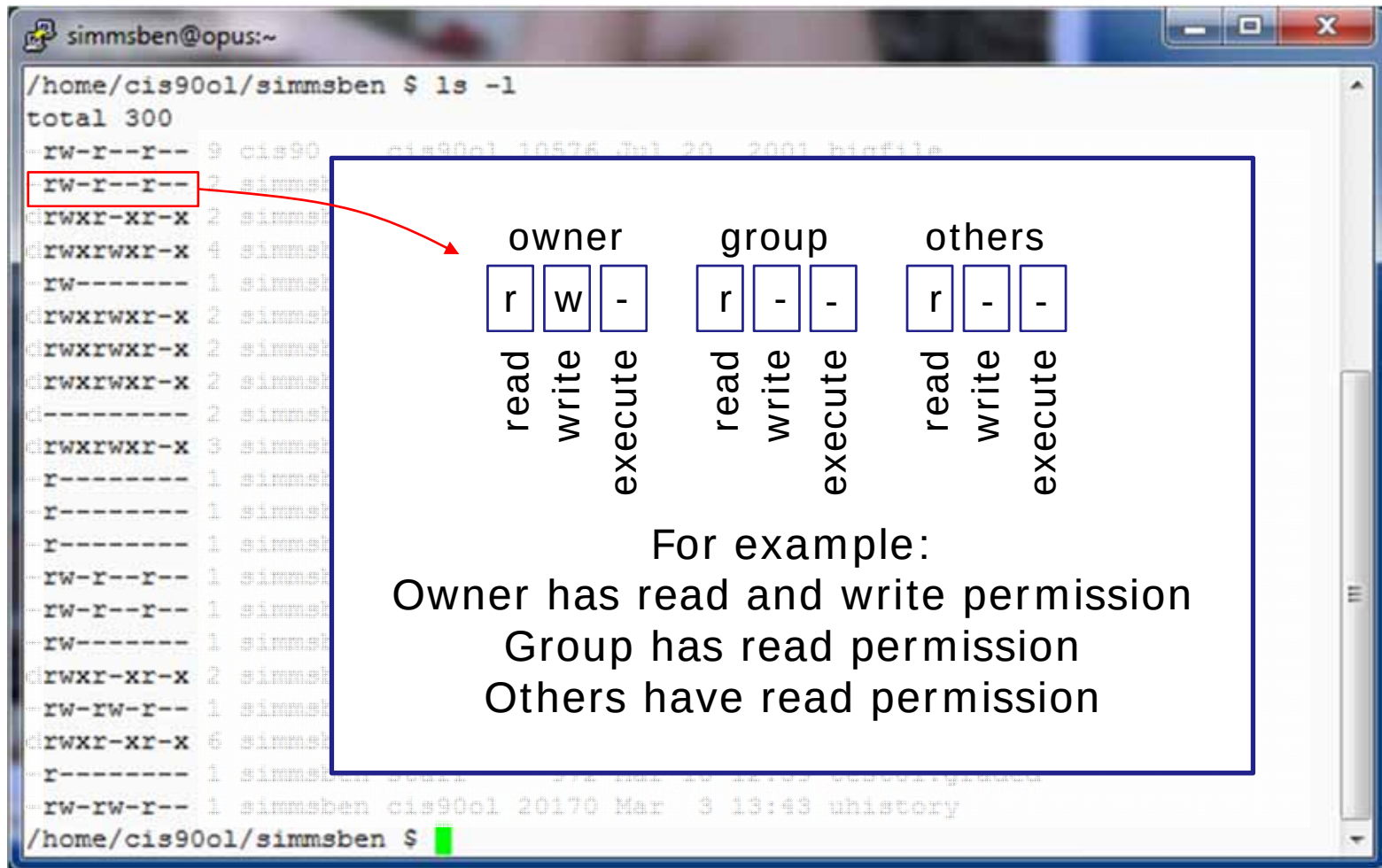
read	write	execute	read	write	execute	read	write	execute

The permission codes are in triplets

Separate permissions apply to the owner, group and all others

File Permissions

Interpreting the permission codes on long listings



The terminal window shows the command `ls -l` being executed in the directory `/home/cis90ol/simmsben`. The output lists several files with their permissions, owner, group, size, date, and name. One line, `-rw-r--r-- 1 simmsben cis90ol 10576 Jul 20 2001 bigfile`, is highlighted with a red box. A red arrow points from this box to a diagram that explains the permission code.

owner			group			others		
r	w	-	r	-	-	r	-	-
read	write	execute	read	write	execute	read	write	execute

For example:
 Owner has read and write permission
 Group has read permission
 Others have read permission

Columns 2-10 show the permissions

File Permissions

Binary

Permissions are stored internally using binary numbers and they can be specified using decimal numbers

Permissions	Binary	Decimal
- - -	000	0
- - X	001	1
- W -	010	2
- W X	011	3
r - -	100	4
r - X	101	5
r W -	110	6
r W X	111	7

For example, rw- (read, write, no execute) = 110 = 6

File Permissions

Binary

If you don't like to think in binary, just remember read=4, write=2 and execute=1 and sum these to get the decimal value of the permission

Permissions	Binary	Decimal
- - -	000	0
- - X	001	1
- W -	010	2
- W X	011	3
r - -	100	4
r - X	101	5
r W -	110	6
r W X	111	7

For example, rw- (4 + 2 + 0) = 6

File Permissions

Interpreting the permission codes on long listings

```

simmsben@opus:~
/home/cis90ol/simmsben $ ls -l
total 300
-rw-r--r--  2 simmsben cis90ol  4096 Mar 17 22:00 class
-rw-r--r--  1 simmsben cis90ol  2947 Mar  3 23:08 dead.letter
drwxrwxr-x  2 simmsben cis90ol  4096 Mar 17 22:05 docs
drwxrwxr-x  2 simmsben cis90ol  4096 Mar 17 22:12 edits
drwxrwxr-x  2 simmsben cis90ol  4096 Mar 17 22:22 etc
drwxrwxr-x  6 simmsben cis90ol  4096 Mar 17 22:20 poems
-rw-rw-r--  1 simmsben staff    972 Mar 18 12:59 test01.graded
-rw-rw-r--  1 simmsben cis90ol 20170 Mar  3 13:43 whistory
/home/cis90ol/simmsben $
  
```

Note, the d in the first column is the file type.
The first column is not part of the permissions.

The permissions on the class directory are?
rwxrwxr-x (775)

- Owner (simmsben) has read, write, execute
- Group (cis90ol) has read, write, execute
- Others have read and execute only

An entry in the long listing of a student home directory

File Permissions

Interpreting the permission codes on long listings

```
simmsben@opus:~
/home/cis90ol/simmsben $ ls -l
total 300
-rw-r--r-- 9 cis90ol cis90ol 10576 Jul 20 2001 bigfile
-rw-r--r-- 2 simmsben cis90ol 10576 Jul 20 2001 bigfile.bak
drwxr-xr-x 2 simmsben cis90ol 4096 Mar 19 16:46 bin
drwxrwxr-x 4 simmsben cis90ol 4096 Mar 17 22:00 class
-rw----- 1 simmsben cis90ol 2947 Mar 3 23:08 dead.letter
drwxrwxr-x 2 simmsben cis90ol 4096 Mar 17 22:05 docs
drwxrwxr-x 2 simmsben cis90ol 4096 Mar 17 22:12 edits
drwxrwxr-x 2 simmsben cis90ol 4096 Mar 17 22:22 etc
d---
drw
-r-
-r-
-r-
-rw
-rw
-rw
drw
-rw
drwxr-xr-x 6 simmsben cis90ol 4096 Mar 17 22:20 poems
-r----- 1 simmsben staff 972 Mar 18 12:59 test01.graded
-rw-rw-r-- 1 simmsben cis90ol 20170 Mar 3 13:43 uhistory
/home/cis90ol/simmsben $
```

The permissions on the dead.letter file are?
rw----- (600)

- Owner (simmsben) has read and write only
- Group (cis90ol) has no permissions
- Others have no permission

Files and directories in /home/cis90ol/simmsben

File Permissions

Interpreting the permission codes on long listings

```
simmsben@opus:~
/home/cis90ol/simmsben $ ls -l
total 300
-rw-r--r-- 9 cis90ol cis90ol 10576 Jul 20 2001 bigfile
-rw-r--r-- 2 simmsben cis90ol 10576 Jul 20 2001 bigfile.bak
drwxr-xr-x 2 simmsben cis90ol 4096 Mar 19 16:46 bin
drwxrwxr-x 4 simmsben cis90ol 4096 Mar 17 22:00 class
-rw----- 1 simmsben cis90ol 2947 Mar 3 23:08 dead.letter
drwxrwxr-x 2 simmsben cis90ol 4096 Mar 17 22:05 docs
drwxrwxr-x 2 simmsben cis90ol 4096 Mar 17 22:12 edits
drwxrwxr-x 2 simmsben cis90ol 4096 Mar 17 22:22 etc
(d...
drwx
-r--
-r--
-r--
-r--
-r--
-rw
drwx
-rw
drwxr-xr-x 6 simmsben cis90ol 4096 Mar 17 22:20 poems
-r----- 1 simmsben staff 972 Mar 18 12:59 test01.graded
-rw-rw-r-- 1 simmsben cis90ol 20170 Mar 3 13:48 whistory
/home/cis90ol/simmsben $
```

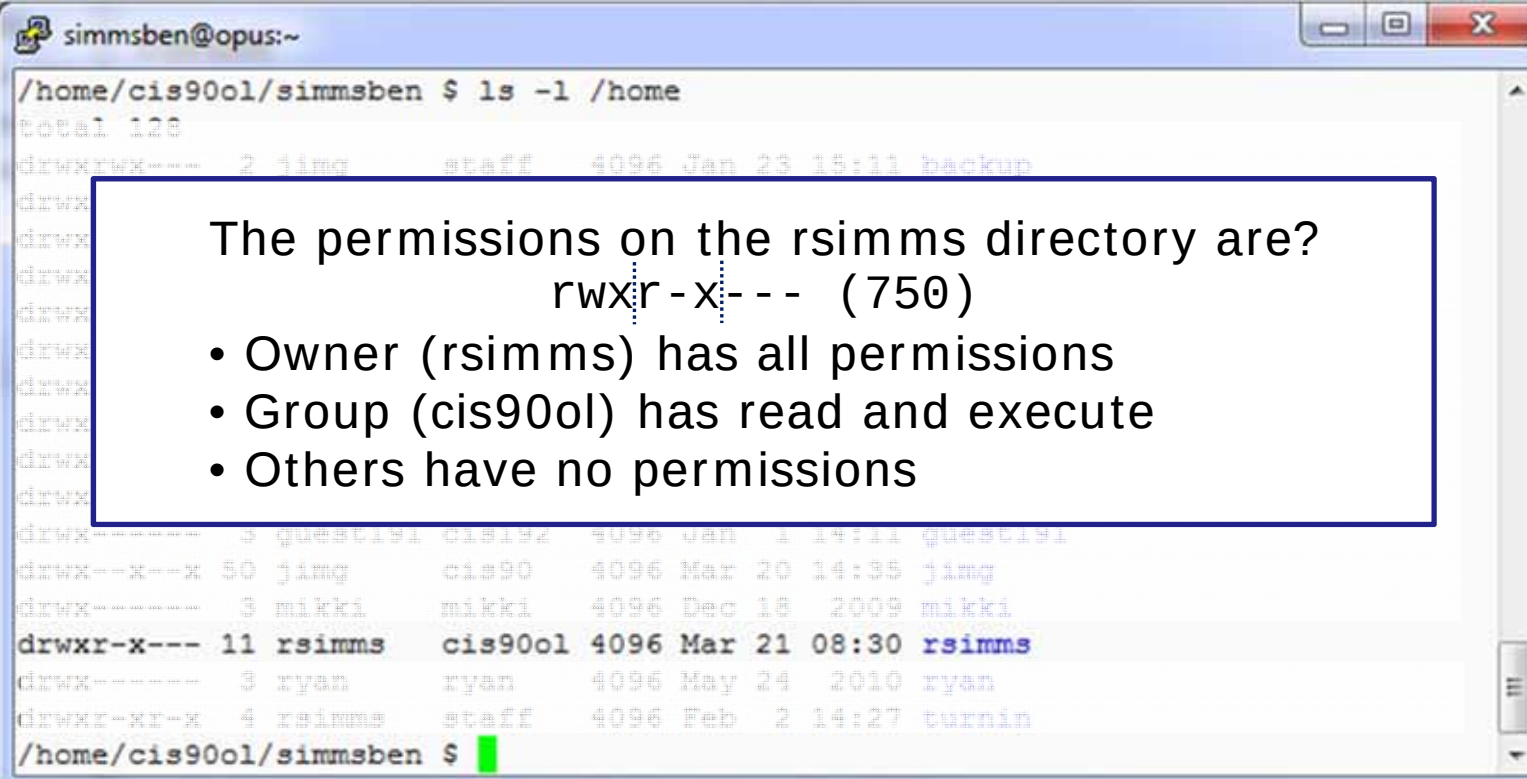
The permissions on the test01.graded file are?
r----- (400)

- Owner (simmsben) has read only
- Group (cis90ol) has no permissions
- Others have no permission

An entry in the long listing of a student home directory

File Permissions

Interpreting the permission codes on long listings



```
simmsben@opus:~  
/home/cis90ol/simmsben $ ls -l /home  
total 120  
drwxrwx--- 2 jimg staff 4096 Jan 23 15:11 backup  
drwxr-x--- 3 rsimms cis90ol 4096 Mar 21 08:30 rsimms  
drwxr-x--- 3 mikki mikki 4096 Dec 18 2009 mikki  
drwxr-x--- 3 ryan ryan 4096 May 24 2010 ryan  
drwxr-xr-x 4 rsimms staff 4096 Feb 2 14:27 turnin  
/home/cis90ol/simmsben $
```

The permissions on the rsimms directory are?

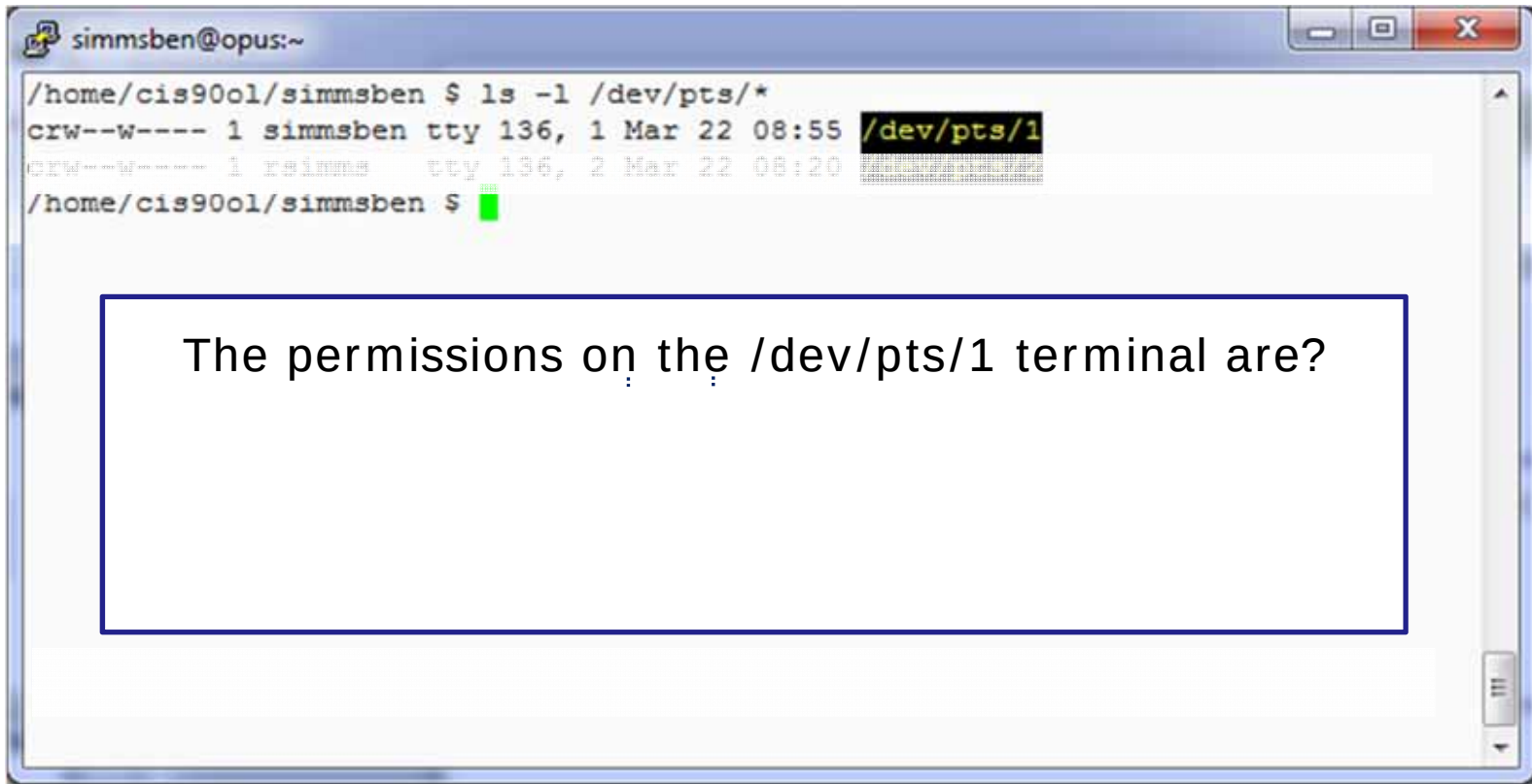
drwxr-x--- (750)

- Owner (rsimms) has all permissions
- Group (cis90ol) has read and execute
- Others have no permissions

An entry in the long listing of the /home directory

File Permissions

Interpreting the permission codes on long listings



A terminal window titled 'simmsben@opus:~' showing the command `ls -l /dev/pts/*` and its output. The output lists two entries: `crw--w---- 1 simmsben tty 136, 1 Mar 22 08:55 /dev/pts/1` and `crw--w---- 1 reimsa tty 136, 2 Mar 22 08:20 /dev/pts/2`. The file `/dev/pts/1` is highlighted in the original image. Below the terminal output, a blue-bordered box contains the question: 'The permissions on the /dev/pts/1 terminal are?'

```
simmsben@opus:~  
/home/cis90ol/simmsben $ ls -l /dev/pts/*  
crw--w---- 1 simmsben tty 136, 1 Mar 22 08:55 /dev/pts/1  
crw--w---- 1 reimsa tty 136, 2 Mar 22 08:20 /dev/pts/2  
/home/cis90ol/simmsben $
```

The permissions on the /dev/pts/1 terminal are?

An entry in the long listing files in the /dev/pts directory

File Permissions

Summary

How do we control access to files and directories?

File Permissions

Summary

What permissions are there?

Answer: **read, write and execute**

File Permissions

Summary

Who do permissions apply to?

Answer:

The **owner** (a user) of the file
The **group** the file belongs to
and everyone else (**others**)

File Permissions

/etc/passwd and /etc/group

```
/home/cis90ol/simmsben $ cat /etc/passwd
< snipped >
simmsben:x:1082:190:Benji Simms:/home/cis90ol/simmsben:/bin/bash
herodbob:x:1083:190:Bo
elmenchr:x:1084:190:Ch
hextcra:x:1085:190:Cra
wilsodan:x:1086:190:Da
mcnamdan:x:1087:190:Da
stumbdav:x:1088:190:Da
roddyduk:x:1089:190:Du
carvaema:x:1090:190:Em
lynbeeri:x:1091:190:Er
vistigab:x:1092:190:Ga
montageo:x:1093:190:Ge
lewisgre:x:1095:190:Gr
wingejas:x:1100:190:Ja
hillejef:x:1101:190:Je
warrejes:x:1102:190:Je
sylvijos:x:1103:190:Jo
cheeken:x:1104:190:Ken
< snipped >
```

```
/home/cis90ol/simmsben $ cat /etc/group
< snipped >
dbus:x:81:
avahi:x:70:
xfs:x:43:
haldaemon:x:68:
gdm:x:42:
avahi-autoipd:x:103:
sabayon:x:86:
apache:x:48:
oprofile:x:16:
audio:x:63:gdm
mikki:x:501:
guest:x:506:
staff:x:503:rsimms,gerlinde,jimg
cis90:x:90:guest,rsimms,jimg
cis130:x:130:jimg
cis164:x:164:
cis172:x:172:
cis90ol:x:190:rsimms
cis191:x:191:rsimms,jimg
cis192:x:192:rsimms
cis193:x:193:
< snipped >
```

*All user accounts are
found in /etc/passwd*

*All groups are
found in
/etc/group*

File Permissions

File owners are users on the system

```
/home/cis90ol/simmsben $ ls -l lab04.graded mbox letter
-r----- 1 simmsben staff      512 Mar 11 14:18 lab04.graded
-rw-r--r-- 1 simmsben cis90ol  1044 Jul 20  2001 letter
-rw----- 1 simmsben cis90ol 99246 Mar  8 10:32 mbox
/home/cis90ol/simmsben $
```

```
/home/cis90ol/simmsben $ id simmsben
uid=1082(simmsben) gid=190(cis90ol) groups=190(cis90ol),100(users)
context=user_u:system_r:unconfined_t
```

*Nice way to select lines
from big files (we will learn
more on **grep** later)*

```
/home/cis90ol/simmsben $ cat /etc/passwd | grep simmsben
simmsben:x:1082:190:Benji Simms:/home/cis90ol/simmsben:/bin/bash
```

*Every user has a unique user ID number (use the **id** command to show) and an entry in the **/etc/passwd** file.*

File Permissions

Files also belong to a group (of users)

The files in this example belong to either the staff or cis900l groups

```
/home/cis900l/simmsben $ ls -l lab04.graded mbox letter
-r----- 1 simmsben staff 512 Mar 11 14:18 lab04.graded
-rw-r--r-- 1 simmsben cis900l 1044 Jul 20 2001 letter
-rw----- 1 simmsben cis900l 99246 Mar 8 10:32 mbox
/home/cis900l/simmsben $
```

```
/home/cis900l/simmsben $ groups simmsben
simmsben : cis900l users
```

The simmsben user is a member of these groups

```
/home/cis900l/simmsben $ groups rsimms
rsimms : staff cis90 cis900l cis191 cis192
```

The rsimms user is a member of these groups

File Permissions

Every user belongs to a primary group (kept in /etc/passwd) and optionally multiple secondary groups

```
/home/cis90ol/simmsben $ id rsimms
uid=201(rsimms) gid=503(staff) ← Primary group
groups=503(staff),90(cis90),190(cis90ol),191(cis191),192(cis192)
context=user_u:system_r:unconfined_t
```

← Secondary groups

```
/home/cis90ol/simmsben $ groups rsimms
rsimms : staff cis90 cis90ol cis191 cis192
```

All groups user belongs to (primary is first group listed)

```
/home/cis90ol/simmsben $ cat /etc/passwd | grep rsimms
rsimms:x:201:503:Rich Simms:/home/rsimms:/bin/bash
```

← Primary group

```
/home/cis90ol/simmsben $ cat /etc/group | grep 503
staff:x:503:rsimms,gerlinde,jing
```

The staff group contains the rsimms user

```
/home/cis90ol/simmsben $ cat /etc/group | grep cis192
cis192:x:192:rsimms
```

The cis192 group contains the rsimms user

Letter file in detail

File Permissions

Relevant fields from the inode

Permissions *Owner* *Group*

```
/home/cis90/simmsben $ ls -l
total 176
drwxrwxr-x 3 simmsben cis90 4096 Mar 18 06:49 africa
-rw-r--r-- 2 simmsben cis90 10576 Jul 20 2001 bigfile
drwxr-xr-x 2 simmsben cis90 4096 Sep 11 2005 bin
-rw-r--r-- 1 simmsben cis90 0 Jul 20 2001 empty
d----- 2 simmsben cis90 4096 Feb 1 2002 Hidden
drwxrwxr-x 4 simmsben cis90 4096 Mar 18 08:12 island
drwxr-xr-x 2 simmsben cis90 4096 Feb 17 2001 Lab2.0
drwxr-xr-x 3 simmsben cis90 4096 Feb 17 2001 Lab2.1
-rw-r--r-- 1 simmsben cis90 1044 Jul 20 2001 letter
drwxr-xr-x 2 simmsben cis90 4096 Sep 11 2005 Miscellaneous
-rw-r--r-- 1 simmsben cis90 759 Jun 6 2002 mission
drwxr-xr-x 5 simmsben cis90 4096 Jan 18 2004 Poems
-rw-r--r-- 1 simmsben cis90 1074 Aug 26 2003 proposal1
-rw-r--r-- 1 simmsben cis90 2175 Jul 20 2001 proposal2
-rw-r--r-- 1 simmsben cis90 2054 Sep 14 2003 proposal3
-rw-rw-r-- 1 simmsben cis90 0 Mar 18 06:36 sawyer
-rw-r--r-- 1 simmsben cis90 1580 Nov 16 2004 small_town
-rw-r--r-- 1 simmsben cis90 485 Aug 26 2003 spellk
-rw-r--r-- 1 simmsben cis90 250 Jul 20 2001 text.err
-rw-r--r-- 1 simmsben cis90 231 Jul 20 2001 text.fxd
-rwxr-xr-x 1 simmsben cis90 509 Jun 6 2002 timecal
-rw-r--r-- 1 simmsben cis90 352 Jul 20 2001 what_am_i
```

File permissions, owners, and groups are displayed in long listings

File Permissions

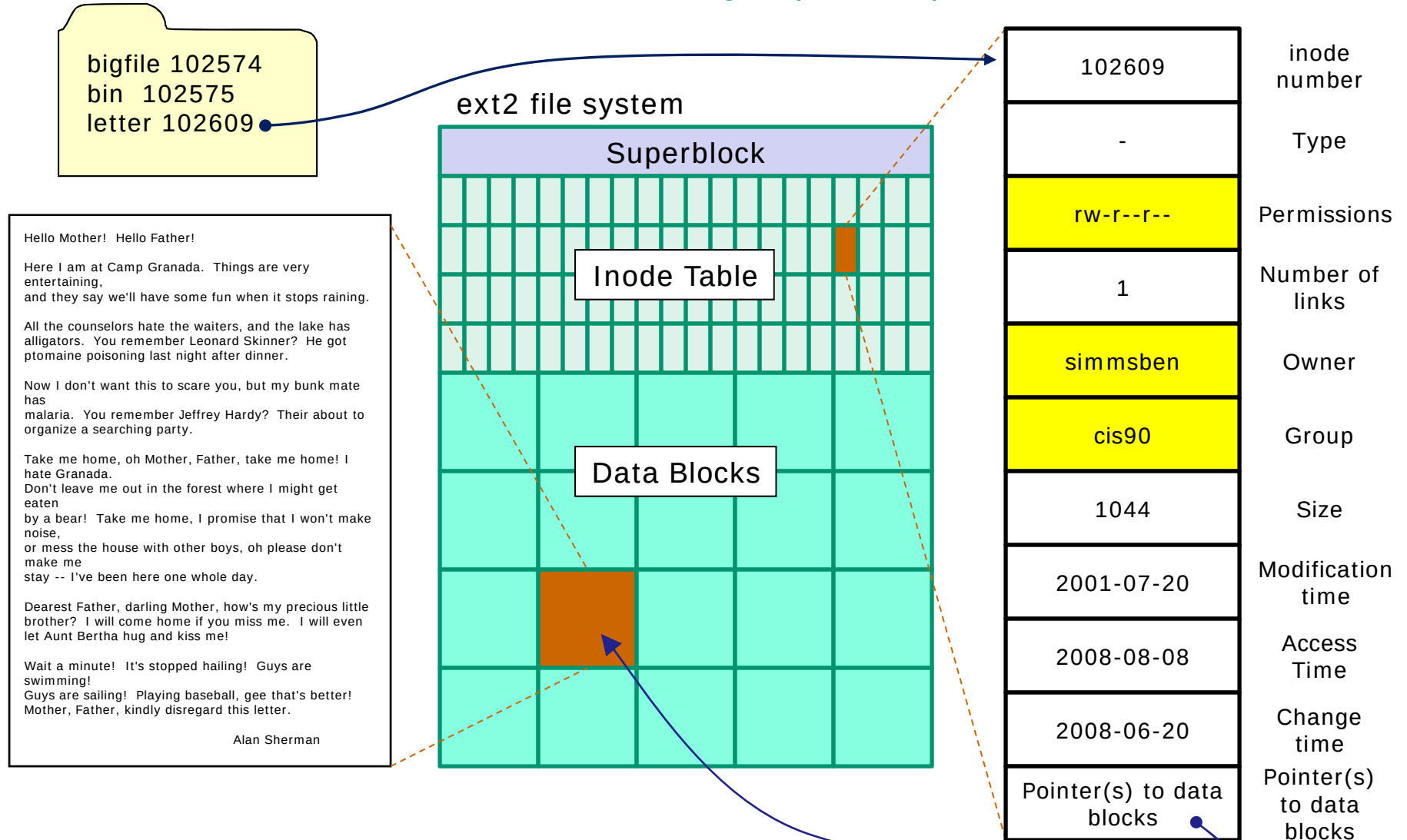
Example: letter file

Permissions *Owner* *Group*

```
/home/cis90/simmsben $ ls -l
total 176
drwxrwxr-x 3 simmsben cis90 4096 Mar 18 06:49 africa
-rw-r--r-- 2 simmsben cis90 10576 Jul 20 2001 bigfile
drwxr-xr-x 2 simmsben cis90 4096 Sep 11 2005 bin
-rw-r--r-- 1 simmsben cis90 0 Jul 20 2001 empty
d----- 2 simmsben cis90 4096 Feb 1 2002 Hidden
drwxrwxr-x 4 simmsben cis90 4096 Mar 18 08:12 island
drwxr-xr-x 2 simmsben cis90 4096 Feb 17 2001 Lab2.0
drwxr-xr-x 3 simmsben cis90 4096 Feb 17 2001 Lab2.1
-rw-r--r-- 1 simmsben cis90 1044 Jul 20 2001 letter
drwxr-xr-x 2 simmsben cis90 4096 Sep 11 2005 Miscellaneous
-rw-r--r-- 1 simmsben cis90 759 Jun 6 2002 mission
drwxr-xr-x 5 simmsben cis90 4096 Jan 18 2004 Poems
-rw-r--r-- 1 simmsben cis90 1074 Aug 26 2003 proposal1
-rw-r--r-- 1 simmsben cis90 2175 Jul 20 2001 proposal2
-rw-r--r-- 1 simmsben cis90 2054 Sep 14 2003 proposal3
-rw-rw-r-- 1 simmsben cis90 0 Mar 18 06:36 sawyer
-rw-r--r-- 1 simmsben cis90 1580 Nov 16 2004 small_town
-rw-r--r-- 1 simmsben cis90 485 Aug 26 2003 spellk
-rw-r--r-- 1 simmsben cis90 250 Jul 20 2001 text.err
-rw-r--r-- 1 simmsben cis90 231 Jul 20 2001 text.fxd
-rwxr-xr-x 1 simmsben cis90 509 Jun 6 2002 timecal
-rw-r--r-- 1 simmsben cis90 352 Jul 20 2001 what_am_i
```

The permissions on letter are rw-r--r-- or 110 100 100 or 644

Permissions, owner and group are kept in the inode of a file



```
[simmsben@opus ~]$ls -il letter
```

```
102609 -rw-r--r-- 1 simmsben cis90 1044 Jul 20 2001 letter
```

File Permissions

Example: letter file

*The **stat** command shows permissions in both formats*

```
[simmsben@opus ~]$ stat letter
  File: `letter'
  Size: 1044          Blocks: 16          IO Block: 4096
    regular file
Device: 805h/2053d    Inode: 102609       Links: 2
Access: (0644/-rw-r--r--)  Uid: ( 1160/simmsben)   Gid: ( 103/
    cis90)
Access: 2008-10-15 14:15:43.000000000 -0700
Modify: 2001-07-20 15:04:39.000000000 -0700
Change: 2008-10-15 14:16:13.000000000 -0700
[simmsben@opus ~]$
```

The permissions on letter are **110100100** or 644 *numeric form*

owner has read and write →
group has only read →
others have only read →

File Permissions

Viewing relevant fields from the inode

The permissions on letter are `rw-r--r--` or 644

owner has read and write

group has only read

others have only read

110100100

110 (binary) = 6 (decimal)

100 (binary) = 4 (decimal)

102609	inode number
-	Type
rw-r--r--	Permissions
1	Number of links
simmsben	Owner
cis90	Group
1044	Size
2001-07-20	Modification time
2008-10-15	Access Time
2008-10-15	Change time
Pointer(s) to data blocks	Pointer(s) to data blocks

More Practice

File Permissions

What is the numeric form of `r--r-----`?

File Permissions

4 4 0
100100000

What is the numeric form of r - - r - - - - ?

Answer: 440

Owner has read

Group has read

Others have no permissions

File Permissions

What is the numeric form of `rw-rw-r--`?

File Permissions

7 6 4
111110100

What is the numeric form of rwxrw-r--?

Answer: 764

File Permissions

What is the numeric form of `rwxr-xr-x`?

File Permissions

7 5 5
111101101

What is the numeric form of `rwxr-xr-x`?

Answer: 755

*Owner has read, write and execute
Group has read and execute
Others have read and execute*

File Permissions

What permissions are 644?

File Permissions

What permissions are 644?

```
110100100  
rw-r--r--
```

Answer:

*owner has read and write
group has read
others have read*

File Permissions

Does the simmsben user have read access to /etc/samba/smb.conf?

File Permissions

Does the simmsben user have read access to /etc/samba/smb.conf?

Answer: yes

```
/home/cis90ol/simmsben $ ls -l /etc/samba/smb.conf  
-rw-r--r-- 1 root root 9733 May 28 2009 /etc/samba/smb.conf
```

root has read & write

root group has read

all other users, including simmsben, have read

Configuring Permissions

File Permissions

Relevant Commands

chown - Changes the ownership of a file. (Only the superuser has this privilege)

chgrp - Changes the group of a file. (Only groups that you belong to)

chmod - Changes the permissions of a file.

- Numeric: **chmod 640 letter** (sets the permissions)
- Mnemonic: **chmod ug+rw letter** (changes the permissions)

❖ Mnemonic: u=user(owner), g=group, o=other
r=read, w=write, x=execute

umask - Sets the default mask applied to permissions for newly created files.

chown

File Permissions

chown and chgrp command - examples

```
[simmsben@opus Directory3]$ touch myfile
[simmsben@opus Directory3]$ ls -l
total 4
-rw-rw-r-- 1 simmsben cis90 0 Oct 15 14:40 myfile
[simmsben@opus Directory3]$ chown rsimms myfile
chown: changing ownership of `myfile': Operation not permitted
```

*Only root can use **chown***

chgrp

File Permissions

chown and chgrp command - examples

```
[simmsben@opus Directory3]$ ls -l
total 4
-rw-rw-r-- 1 simmsben cis90 0 Oct 15 14:40 myfile
```

```
[simmsben@opus Directory3]$ chgrp users myfile           change group to users
[simmsben@opus Directory3]$ ls -l
total 4
-rw-rw-r-- 1 simmsben users 0 Oct 15 14:40 myfile
```

```
[simmsben@opus Directory3]$ chgrp cis90 myfile           change group back to cis90
[simmsben@opus Directory3]$ ls -l
total 4
-rw-rw-r-- 1 simmsben cis90 0 Oct 15 14:40 myfile
```

*Use **chgrp** to change the group a file belongs to*

chmod

File Permissions

chmod command - examples

```
/home/cis90/simmsben/Directory3 $ ls -l myfile  
-rw-rw-r-- 1 simmsben cis90 0 Oct 13 07:16 myfile
```

With chmod command use “u” for owner (user), “g” for group and “o” for others

```
/home/cis90/simmsben/Directory3 $ chmod u+x myfile      add execute  
/home/cis90/simmsben/Directory3 $ ls -l myfile           permission for  
-rwxrw-r-- 1 simmsben cis90 0 Oct 13 07:16 myfile        owner
```

```
/home/cis90/simmsben/Directory3 $ chmod g+x myfile      add execute  
/home/cis90/simmsben/Directory3 $ ls -l myfile           permission for  
-rwxrwxr-- 1 simmsben cis90 0 Oct 13 07:16 myfile        group
```

Use chmod to add or remove permissions from a file

File Permissions

chmod command - examples

```
[simmsben@opus Directory3]$ ls -l myfile  
-rwxrwxr-- 1 simmsben cis90 0 Oct 15 14:40 myfile
```

```
[simmsben@opus Directory3]$ chmod -x myfile
```

remove execute from all

```
[simmsben@opus Directory3]$ ls -l myfile  
-rw-rw-r-- 1 simmsben cis90 0 Oct 15 14:40 myfile
```

```
[simmsben@opus Directory3]$ chmod og+x myfile
```

*add execute to
others and group*

```
[simmsben@opus Directory3]$ ls -l myfile  
-rw-rwxr-x 1 simmsben cis90 0 Oct 15 14:40 myfile
```

```
[simmsben@opus Directory3]$ chmod go-rwx myfile
```

*remove read,
write, execute
from groups and
others*

```
[simmsben@opus Directory3]$ ls -l myfile  
-rw----- 1 simmsben cis90 0 Oct 15 14:40 myfile
```

Use chmod to add or remove permissions from a file

File Permissions

chmod command - examples

```
[simmsben@opus Directory3]$ chmod 664 myfile  
[simmsben@opus Directory3]$ ls -l myfile  
-rw-rw-r-- 1 simmsben cis90 0 Oct 15 14:40 myfile
```

*You can also specify each permission directly
using the numeric mode of the command*

File Permissions

chmod command - examples

```
[simmsben@opus Directory3]$ chmod 777 myfile  
[simmsben@opus Directory3]$ ls -l myfile  
-rw-rw-rwx 1 simmsben cis90 0 Oct 15 14:40 myfile
```

```
[simmsben@opus Directory3]$ chmod 640 myfile  
[simmsben@opus Directory3]$ ls -l myfile  
-rw-r----- 1 simmsben cis90 0 Oct 15 14:40 myfile
```

```
[simmsben@opus Directory3]$ chmod 000 myfile  
[simmsben@opus Directory3]$ ls -l myfile  
------ 1 simmsben cis90 0 Oct 15 14:40 myfile
```

```
[simmsben@opus Directory3]$ chmod 644 myfile  
[simmsben@opus Directory3]$ ls -l myfile  
-rw-r--r-- 1 simmsben cis90 0 Oct 15 14:40 myfile
```

*More examples using the numeric mode of the **chmod** command*

File Permissions

Commands that require file permissions

Permission	File	Directory
Read (4)	cat, more, head, tail, cp	ls
Write (2)	vi, saving mail	cp, mv, rm, ln
Execute (1)	\$ command	cd, ls -l, find

To use the cat, more, file, head, tail and cp (source file) commands on a file it is necessary to have read permission

Read Permission

Make a directory named Directory3, cd into it, and create myfile:

```
/home/cis90/simmsben $ mkdir Directory3
/home/cis90/simmsben $ cd Directory3/
/home/cis90/simmsben/Directory3 $ touch myfile
/home/cis90/simmsben/Directory3 $ ls -l myfile
-rw-r--r-- 1 simmsben cis90 0 Oct 13 07:16 myfile
```

Add some data to myfile and try reading with and without read permission:

```
/home/cis90/simmsben/Directory3 $ echo Blah Blah Blah > myfile
/home/cis90/simmsben/Directory3 $ cat myfile
Blah Blah Blah
/home/cis90/simmsben/Directory3 $ chmod u-r myfile
/home/cis90/simmsben/Directory3 $ ls -l myfile
--w-r--r-- 1 simmsben cis90 15 Oct 13 08:50 myfile
/home/cis90/simmsben/Directory3 $ cat myfile
cat: myfile: Permission denied
```

removes read permission for user owning the file

Can you fix this so you can read your own file again?

File Permissions

Commands that require file permissions

Permission	File	Directory
Read (4)	cat, more, head, tail, cp	ls
Write (2)	vi, saving mail	cp, mv, rm, ln
Execute (1)	\$ command	cd, ls -l, find

Write Permission

Start with a fresh version of myfile:

```
/home/cis90/simmsben/Directory3 $ rm myfile  
/home/cis90/simmsben/Directory3 $ touch myfile  
/home/cis90/simmsben/Directory3 $ ls -l myfile  
-rw-rw-r-- 1 simmsben cis90 0 Oct 13 08:58 myfile
```

Add some data to myfile :

```
/home/cis90/simmsben/Directory3 $ echo Blah Blah Blah > myfile  
/home/cis90/simmsben/Directory3 $ chmod 444 myfile write permission removed  
/home/cis90/simmsben/Directory3 $ ls -l myfile  
-r--r--r-- 1 simmsben cis90 15 Oct 13 09:02 myfile  
/home/cis90/simmsben/Directory3 $ echo Blah Blah Blah > myfile  
-bash: myfile: Permission denied
```

Can you fix this so you can write to your own file again?

File Permissions

Commands that require file permissions

Permission	File	Directory
Read (4)	cat, more, head, tail, cp	ls
Write (2)	vi, saving mail	cp, mv, rm, ln
Execute (1)	\$ command	cd, ls -l, find



Execute Permission

Start with a fresh version of myfile:

```
/home/cis90/simmsben/Directory3 $ rm myfile
rm: remove write-protected regular file `myfile'? yes
/home/cis90/simmsben/Directory3 $ touch myfile
/home/cis90/simmsben/Directory3 $ ls -l myfile
-rw-rw-r-- 1 simmsben cis90 0 Oct 13 09:12 myfile
```

Make a little script and give it execute permission:

```
/home/cis90/simmsben/Directory3 $ echo 'banner $LOGNAME is cool' > myfile
/home/cis90/simmsben/Directory3 $ cat myfile
banner $LOGNAME is cool
/home/cis90/simmsben/Directory3 $ myfile
-bash: ./myfile: Permission denied
/home/cis90/simmsben/Directory3 $ chmod +x myfile
/home/cis90/simmsben/Directory3 $ ls -l myfile
-rwxrwxr-x 1 simmsben cis90 24 Oct 13 09:27 myfile
/home/cis90/simmsben/Directory3 $ myfile
```

*add execute permission
for all users*

What happens now when you type myfile?

removing a file example 1

File Permissions

Commands that require file permissions

Directories contain filename and inode number pairs

Permission	File	Directory
Read (4)	cat, more, head, tail, cp	ls
Write (2)	vi, saving mail	cp, mv, rm ln
Execute (1)	\$ command	cd, ls -l, find

To remove a file, you must have write permission on the directory containing it! That's because you are writing an updated list of filename and inode pairs to the directory (that no longer contain the removed file).

File permissions and rm command example 1

```
[simmsben@opus ~]$ ls -ld Directory3  
dr-xrwxr-x 2 simmsben cis90 4096 Oct 15 15:00 Directory3
```

```
[simmsben@opus ~]$ cd Directory3  
[simmsben@opus Directory3]$ ls -l myfile  
-rw-r--r-- 1 simmsben cis90 0 Oct 15 15:00 myfile
```

*Permissions on myfile
are 644. Owner has
read and write
permission.*

```
[simmsben@opus Directory3]$ rm myfile  
rm: cannot remove `myfile': Permission denied
```

```
[simmsben@opus Directory3]$ chmod 777 myfile  
[simmsben@opus Directory3]$ ls -l myfile  
-rwxrwxrwx 1 simmsben cis90 0 Oct 15 15:00 myfile
```

*Permissions on myfile
are 777. Everyone
has all permissions.*

```
[simmsben@opus Directory3]$ rm myfile  
rm: cannot remove `myfile': Permission denied
```

So why can't Benji remove his own file?



Answer: Removing a file requires write permission on the directory that contains the file. The permissions on the file itself do not apply.

```
[simmsben@opus ~]$ ls -ld Directory3  
dr-xrwxr-x 2 simmsben cis90 4096 Oct 15 15:00 Directory3
```



Without write permission, Benji cannot remove any files from this directory

removing a file example 2

File Permissions

Commands that require file permissions

Directories contain filename and inode number pairs

Permission	File	Directory
Read (4)	cat, more, head, tail, cp	ls
Write (2)	vi, saving mail	cp, mv, rm ln
Execute (1)	\$ command	cd, ls -l, find

To remove a file, you must have write permission on the directory containing it! That's because you are writing an updated list of filename and inode pairs to the directory (that no longer contain the removed file).

File permissions and rm command

Example 2

```
[simmsben@opus ~]$ ls -ld Directory3  
drwxr-xr-x 2 simmsben cis90 4096 Oct 15 15:00 Directory3
```

```
[simmsben@opus ~]$ cd Directory3  
[simmsben@opus Directory3]$ chmod 000 myfile  
[simmsben@opus Directory3]$ ls -l myfile  
----- 1 simmsben cis90 0 Oct 15 15:00 myfile
```

*Now Beni has no
permissions on
this file*

```
[simmsben@opus Directory3]$ rm myfile  
rm: remove write-protected regular empty file `myfile'? yes  
[simmsben@opus Directory3]$
```

So how come he can delete it?



Answer: Removing a file requires write permission on the directory that contains the file. The permissions on the file itself do not apply.

```
[simmsben@opus ~]$ ls -ld Directory3  
drwxr-xr-x 2 simmsben cis90 4096 Oct 15 15:00 Directory3
```



With write permission, Benji can remove any of the files from this directory ... even the ones he does not have read & write permission for.

Directories contain filename and inode pairs. Removing a file deletes the pair from the directory. This requires write permission on the directory, not the file.

bigfile 102574
bin 102575
letter 102609

ext2 file system

Superblock

Inode Table

Data Blocks

Hello Mother! Hello Father!

Here I am at Camp Granada. Things are very entertaining,
and they say we'll have some fun when it stops raining.

All the counselors hate the waiters, and the lake has alligators. You remember Leonard Skinner? He got ptomaine poisoning last night after dinner.

Now I don't want this to scare you, but my bunk mate has malaria. You remember Jeffrey Hardy? Their about to organize a searching party.

Take me home, oh Mother, Father, take me home! I hate Granada.
Don't leave me out in the forest where I might get eaten
by a bear! Take me home, I promise that I won't make noise,
or mess the house with other boys, oh please don't make me
stay -- I've been here one whole day.

Dearest Father, darling Mother, how's my precious little brother? I will come home if you miss me. I will even let Aunt Bertha hug and kiss me!

Wait a minute! It's stopped hailing! Guys are swimming!
Guys are sailing! Playing baseball, gee that's better!
Mother, Father, kindly disregard this letter.

Alan Sherman

102609	inode number
-	Type
rw-r--r--	Permissions
1	Number of links
simmsben	Owner
cis90	Group
1044	Size
2001-07-20	Modification time
2008-08-08	Access Time
2008-06-20	Change time
Pointer(s) to data blocks	Pointer(s) to data blocks

96

```
[simmsben@opus ~]$ls -il letter
102609 -rw-r--r-- 1 simmsben cis90 1044 Jul 20 2001 letter
```

umask

File Permissions

Default Permissions

What permissions are files created with?

Answer: This is determined by the umask setting

Default permissions

- Default permissions for an ordinary file: `rw-rw-rw-` `666`
- Default permissions for directories: `rwxrwxrwx` `777`

The umask is a three digit octal value whose bits strip away (mask off) default permissions:

- umask 777 - strips off all permissions from a file or directory
- umask 000 - leaves the default permissions alone
- umask 022 - strips off write permissions from group and other

File Permissions

umask - examples

```
[simmsben@opus Directory3]$ umask          This displays current umask setting
0002
[simmsben@opus Directory3]$ rm myfile
[simmsben@opus Directory3]$ touch myfile
[simmsben@opus Directory3]$ ls -l
total 4
-rw-rw-r-- 1 simmsben cis90 0 Oct 15 14:59 myfile
```

666	rw-rw-rw-	<i>default permissions for a file</i>
002	-----w-	<i>umask value (strips these permissions from default)</i>
664	rw-rw-r--	<i>result after masking</i>

File Permissions

umask - examples

```
[simmsben@opus Directory3]$ umask 000           Change umask to 000
[simmsben@opus Directory3]$ rm myfile
[simmsben@opus Directory3]$ touch myfile
[simmsben@opus Directory3]$ ls -l
total 4
-rw-rw-rw- 1 simmsben cis90 0 Oct 15 15:00 myfile
```

666	rw-rw-rw-	<i>default permissions for a file</i>
000	-----	<i>umask value (strips these permissions from default)</i>
666	rw-rw-rw-	<i>result after masking</i>

File Permissions

umask - examples

```
[simmsben@opus Directory3]$ umask 022 Change umask to 022
[simmsben@opus Directory3]$ rm myfile
[simmsben@opus Directory3]$ touch myfile
[simmsben@opus Directory3]$ ls -l
total 4
-rw-r--r-- 1 simmsben cis90 0 Oct 15 15:00 myfile
```

666	rw-rw-rw-	<i>default permissions for a file</i>
022	- - - -w - -w -	<i>umask value (strips these permissions from default)</i>
644	rw-r--r--	<i>result after masking</i>

Examples setup

File Permissions

exercise - setup



```
/home/cis90/simmsben $ mkdir Directory1 Directory2
/home/cis90/simmsben $ chmod 751 Directory1
/home/cis90/simmsben $ chmod 775 Directory2
/home/cis90/simmsben $ ls -ld Directory*
drwxr-x--x 2 simmsben cis90 4096 Oct 15 11:11 Directory1
drwxrwxr-x 2 simmsben cis90 4096 Oct 15 11:11 Directory2
```

```
/home/cis90/simmsben $ cd Directory1
```

```
/home/cis90/simmsben/Directory1 $ echo "blah blah blah" > file1
/home/cis90/simmsben/Directory1 $ echo "blah blah blah" > file2
/home/cis90/simmsben/Directory1 $ echo "blah blah blah" > file3
```

```
/home/cis90/simmsben/Directory1 $ chmod 664 file1
/home/cis90/simmsben/Directory1 $ chmod 755 file2
/home/cis90/simmsben/Directory1 $ chmod 554 file3
```

```
/home/cis90/simmsben/Directory1 $ ls -l
total 24
-rw-rw-r-- 1 simmsben cis90 15 Oct 15 11:13 file1
-rwxr-xr-x 1 simmsben cis90 15 Oct 15 11:13 file2
-r-xr-xr-- 1 simmsben cis90 15 Oct 15 11:13 file3
```

File Permissions

exercise - setup



```
[simmsben@opus ~]$ ls -ld /home /home/cis90 /home/cis90/simmsben
drwxr-xr-x 16 root      root   4096 Jan 27 19:20 /home
drwxr-x--- 33 guest90   cis90 4096 Feb 19 05:49 /home/cis90
drwxr-xr-x 13 simmsben  cis90 4096 Mar 30 06:47 /home/cis90/simmsben
```

Note that other users have no permissions for the /home/cis90 directory

Examples owner write to files

File Permissions

exercise – can **owner** change these files?

Can Benji write to file1?



```
/home/cis90/simmsben/Directory1 $ ls -l
total 24
-rw-rw-r-- 1 simmsben cis90 15 Oct 15 11:13 file1
-rwxr-xr-x 1 simmsben cis90 15 Oct 15 11:13 file2
-r-xr-xr-- 1 simmsben cis90 15 Oct 15 11:13 file3
/home/cis90/simmsben/Directory1 $
```

```
/home/cis90/simmsben/Directory1 $ echo "changes" > file1
```



File Permissions

exercise – can **owner** change these files?

Yes, he can. Can Benji write to file2?



```
/home/cis90/simmsben/Directory1 $ ls -l
total 24
-rw-rw-r-- 1 simmsben cis90 15 Oct 15 11:13 file1
-rwxr-xr-x 1 simmsben cis90 15 Oct 15 11:13 file2
-r-xr-xr-- 1 simmsben cis90 15 Oct 15 11:13 file3
/home/cis90/simmsben/Directory1 $
```

```
/home/cis90/simmsben/Directory1 $ echo "changes" > file1 
/home/cis90/simmsben/Directory1 $ echo "changes" > file2
```



File Permissions

exercise – can **owner** change these files?

Yes, he can. Can Benji write to file3?



```
/home/cis90/simmsben/Directory1 $ ls -l
total 24
-rw-rw-r-- 1 simmsben cis90 15 Oct 15 11:13 file1
-rwxr-xr-x 1 simmsben cis90 15 Oct 15 11:13 file2
-r-xr-xr-- 1 simmsben cis90 15 Oct 15 11:13 file3
/home/cis90/simmsben/Directory1 $
```

```
/home/cis90/simmsben/Directory1 $ echo "changes" > file1 ✓
/home/cis90/simmsben/Directory1 $ echo "changes" > file2 ✓
/home/cis90/simmsben/Directory1 $ echo "changes" > file3
```



File Permissions

exercise – can **owner** change these files?

No he cannot



```
/home/cis90/simmsben/Directory1 $ ls -l
total 24
-rw-rw-r-- 1 simmsben cis90 15 Oct 15 11:13 file1
-rwxr-xr-x 1 simmsben cis90 15 Oct 15 11:13 file2
-r-xr-xr-- 1 simmsben cis90 15 Oct 15 11:13 file3
/home/cis90/simmsben/Directory1 $
```

```
/home/cis90/simmsben/Directory1 $ echo "changes" > file1
/home/cis90/simmsben/Directory1 $ echo "changes" > file2
/home/cis90/simmsben/Directory1 $ echo "changes" > file3
-bash: file3: Permission denied
/home/cis90/simmsben/Directory1 $
```



```
/home/cis90/simmsben/Directory1 $ cat f*
changes
changes
blah blah blah
/home/cis90/simmsben/Directory1 $
```

Benji could write to file1 and file2 but not file3

Examples group write to files

File Permissions

exercise – can **group user** change these files?

Can Duke write to Benji's file1?



```
[roddyduk@opus Directory1]$ ls -l
total 24
-rw-rw-r-- 1 simmsben cis90  8 Oct 15 11:48 file1
-rwxr-xr-x 1 simmsben cis90 15 Oct 15 11:13 file2
-r-xr-xr-- 1 simmsben cis90 15 Oct 15 11:13 file3
[roddyduk@opus Directory1]$ echo "changes" > file1
```



File Permissions

exercise – can **group user** change these files?

Yes, he can. Can Duke write to Benji's file2?



```
[roddyduk@opus Directory1]$ ls -l
total 24
-rw-rw-r-- 1 simmsben cis90  8 Oct 15 11:48 file1
-rwxr-xr-x 1 simmsben cis90 15 Oct 15 11:13 file2
-r-xr-xr-- 1 simmsben cis90 15 Oct 15 11:13 file3
[roddyduk@opus Directory1]$ echo "changes" > file1
[roddyduk@opus Directory1]$ echo "changes" > file2
```



File Permissions

exercise – can **group user** change these files?

No, he cannot. Can Duke write to Benji's file3?



```
[roddyduk@opus Directory1]$ ls -l
total 24
-rw-rw-r-- 1 simmsben cis90  8 Oct 15 11:48 file1
-rwxr-xr-x 1 simmsben cis90 15 Oct 15 11:13 file2
-r-xr-xr-- 1 simmsben cis90 15 Oct 15 11:13 file3
[roddyduk@opus Directory1]$ echo "changes" > file1
[roddyduk@opus Directory1]$ echo "changes" > file2
-bash: file2: Permission denied
[roddyduk@opus Directory1]$ echo "changes" > file3
```



File Permissions

exercise – can **group user** change these files?

No, he cannot.



```
[roddyduk@opus Directory1]$ ls -l
total 24
-rw-rw-r-- 1 simmsben cis90  8 Oct 15 11:48 file1
-rwxr-xr-x 1 simmsben cis90 15 Oct 15 11:13 file2
-r-xr-xr-- 1 simmsben cis90 15 Oct 15 11:13 file3
[roddyduk@opus Directory1]$ echo "changes" > file1
[roddyduk@opus Directory1]$ echo "changes" > file2
-bash: file2: Permission denied
[roddyduk@opus Directory1]$ echo "changes" > file3
-bash: file3: Permission denied
[roddyduk@opus Directory1]$
```



Examples other write to files

File Permissions



exercise – can **other user** change these files?

```
[simmsben@opus ~]$ ls -ld /home /home/cis90 /home/cis90/simmsben
drwxr-xr-x 16 root      root   4096 Jan 27 19:20 /home
drwxr-x--- 33 rsimms   cis90  4096 Feb 19 05:49 /home/cis90
drwxr-xr-x 13 simmsben cis90  4096 Mar 30 06:47 /home/cis90/simmsben
[simmsben@opus Directory1]$ cd; ls -l D*1/file*
-rw-rw-r-- 1 simmsben cis90 15 Mar 30 06:49 Directory1/file1
-rwxr-xr-x 1 simmsben cis90 15 Mar 30 06:49 Directory1/file2
-r-xr-xr-- 1 simmsben cis90 15 Mar 30 06:49 Directory1/file3
```



Mary is not in the cis90 group. Can she write to any of Benji's files?

File Permissions



exercise – can **other user** change these files?

```
[simmsben@opus ~]$ ls -ld /home /home/cis90 /home/cis90/simmsben
drwxr-xr-x 16 root      root   4096 Jan 27 19:20 /home
drwxr-x--- 33 rsimms    cis90  4096 Feb 19 05:49 /home/cis90
drwxr-xr-x 13 simmsben  cis90  4096 Mar 30 06:47 /home/cis90/simmsben
[simmsben@opus Directory1]$ cd; ls -l D*1/file*
-rw-rw-r-- 1 simmsben cis90 15 Mar 30 06:49 Directory1/file1
-rwxr-xr-x 1 simmsben cis90 15 Mar 30 06:49 Directory1/file2
-r-xr-xr-- 1 simmsben cis90 15 Mar 30 06:49 Directory1/file3
```



Mary is not in the cis90 group. She does not have read permission to /home/cis90 or write permission to any of Benji's files

```
[simmsmar@opus ~]$ ls -l /home/cis90/simmsben/Directory1 ❌
ls: /home/cis90/simmsben/Directory1: Permission denied
[simmsmar@opus ~]$ echo "simmsmar" > /home/cis90/simmsben/Directory1/file1 ❌
-bash: /home/cis90/simmsben/Directory1/file1: Permission denied
[simmsmar@opus ~]$ echo "simmsmar" > /home/cis90/simmsben/Directory1/file2 ❌
-bash: /home/cis90/simmsben/Directory1/file2: Permission denied
[simmsmar@opus ~]$ echo "simmsmar" > /home/cis90/simmsben/Directory1/file3 ❌
-bash: /home/cis90/simmsben/Directory1/file3: Permission denied
[simmsmar@opus ~]$
```

Examples owner remove files

File Permissions

exercise – can the **owner** remove these files?



```
[simmsben@opus Directory1]$ ls -ld
drwxr-x--x 2 simmsben cis90 4096 Oct 15 12:42 .
[simmsben@opus Directory1]$
[simmsben@opus Directory1]$ ls -l
total 24
-rw-rw-r-- 1 simmsben cis90 15 Oct 15 11:13 file1
-rwxr-xr-x 1 simmsben cis90 15 Oct 15 11:13 file2
-r-xr-xr-- 1 simmsben cis90 15 Oct 15 11:13 file3
```

Can Benji remove his own files?

File Permissions

exercise – can the **owner** remove these files?

Can Benji remove his files?



```
[simmsben@opus Directory1]$ ls -ld
drwxr-x--x 2 simmsben cis90 4096 Oct 15 12:42 .
[simmsben@opus Directory1]$
[simmsben@opus Directory1]$ ls -l
total 24
-rw-rw-r-- 1 simmsben cis90 15 Oct 15 11:13 file1
-rwxr-xr-x 1 simmsben cis90 15 Oct 15 11:13 file2
-r-xr-xr-- 1 simmsben cis90 15 Oct 15 11:13 file3
[simmsben@opus Directory1]$ rm file1
[simmsben@opus Directory1]$ rm file2
[simmsben@opus Directory1]$ rm file3
rm: remove write-protected regular file `file3'? y
[simmsben@opus Directory1]$
```



Yes he can, even the one he does not have write permission to!

Examples

group

remove files

File Permissions

exercise – can **group user** remove these files?



```
[simmsben@opus Directory1]$ ls -ld
drwxr-x--x 2 simmsben cis90 4096 Oct 15 12:42 .
```

```
[simmsben@opus Directory1]$ ls -l
total 24
-rw-rw-r-- 1 simmsben cis90 15 Oct 15 11:13 file1
-rwxr-xr-x 1 simmsben cis90 15 Oct 15 11:13 file2
-r-xr-xr-- 1 simmsben cis90 15 Oct 15 11:13 file3
```



Can Duke remove Benji's files?

File Permissions

exercise – can **group user** remove these files?



Can Duke remove Benji's files?

```
[simmsben@opus Directory1]$ ls -ld
drwxr-x--x 2 simmsben cis90 4096 Oct 15 12:42 .
```

```
[simmsben@opus Directory1]$ ls -l
total 24
-rw-rw-r-- 1 simmsben cis90 15 Oct 15 11:13 file1
-rwxr-xr-x 1 simmsben cis90 15 Oct 15 11:13 file2
-r-xr-xr-- 1 simmsben cis90 15 Oct 15 11:13 file3
```



Nope, he does not have write permission to Benji's directory

```
[roddyduk@opus ~]$ rm ../simmsben/Directory1/file1
rm: cannot remove '../simmsben/Directory1/file1': Permission denied
[roddyduk@opus ~]$ rm ../simmsben/Directory1/file2
rm: remove write-protected regular file '../simmsben/Directory1/file2'? y
rm: cannot remove '../simmsben/Directory1/file2': Permission denied
[roddyduk@opus ~]$ rm ../simmsben/Directory1/file3
rm: remove write-protected regular file '../simmsben/Directory1/file3'? y
rm: cannot remove '../simmsben/Directory1/file3': Permission denied
[roddyduk@opus ~]$
```



Examples other remove files

File Permissions

exercise – can **other user** remove these files?



```
[simmsben@opus ~]$ ls -ld /home /home/cis90 /home/cis90/simmsben
drwxr-xr-x 16 root      root  4096 Jan 27 19:20 /home
drwxr-x--- 33 rsimms    cis90 4096 Feb 19 05:49 /home/cis90
drwxr-xr-x 13 simmsben cis90 4096 Mar 30 06:47 /home/cis90/simmsben
```

```
[simmsben@opus Directory1]$ ls -l
total 24
-rw-rw-r-- 1 simmsben cis90 15 Oct 15 11:13 file1
-rwxr-xr-x 1 simmsben cis90 15 Oct 15 11:13 file2
-r-xr-xr-- 1 simmsben cis90 15 Oct 15 11:13 file3
```



Can Mary remove Benji's files?

File Permissions

exercise – can **other user** remove these files?



Can Mary remove Benji's files?

```
[simmsben@opus /]$ ls -ld /home /home/cis90 /home/cis90/simmsben
drwxr-xr-x 16 root      root   4096 Jan 27 19:20 /home
drwxr-x--- 33 rsimms    cis90  4096 Feb 19 05:49 /home/cis90
drwxr-xr-x 13 simmsben cis90  4096 Mar 30 06:47 /home/cis90/simmsben
```

```
[simmsben@opus Directory1]$ ls -l
total 24
-rw-rw-r-- 1 simmsben cis90 15 Oct 15 11:13 file1
-rwxr-xr-x 1 simmsben cis90 15 Oct 15 11:13 file2
-r-xr-xr-- 1 simmsben cis90 15 Oct 15 11:13 file3
```



*Nope, no read permission on **/home/cis90** or write permission on **/home/cis90/simmsben***

```
[simmsmar@opus ~]$ rm /home/cis90/simmsben/Directory1/file1
rm: cannot remove `/home/cis90/simmsben/Directory1/file1': Permission denied ❌
[simmsmar@opus ~]$ rm /home/cis90/simmsben/Directory1/file2
rm: cannot remove `/home/cis90/simmsben/Directory1/file2': Permission denied ❌
[simmsmar@opus ~]$ rm /home/cis90/simmsben/Directory1/file3
rm: cannot remove `/home/cis90/simmsben/Directory1/file3': Permission denied ❌
[simmsmar@opus ~]$
```

Examples owner read files

File Permissions

exercise – can **owner** read these files?



```
[simmsben@opus Directory1]$ ls -ld
drwxr-x--x 2 simmsben cis90 4096 Oct 15 12:42 .
[simmsben@opus Directory1]$
[simmsben@opus Directory1]$ ls -l
total 24
-rw-rw-r-- 1 simmsben cis90 15 Oct 15 11:13 file1
-rwxr-xr-x 1 simmsben cis90 15 Oct 15 11:13 file2
-r-xr-xr-- 1 simmsben cis90 15 Oct 15 11:13 file3
```

Can Benj read his own files?

File Permissions

exercise – can **owner** read these files?

Can Benj read his own files?



```
[simmsben@opus Directory1]$ ls -ld
drwxr-x--x 2 simmsben cis90 4096 Oct 15 12:42 .
[simmsben@opus Directory1]$
[simmsben@opus Directory1]$ ls -l
total 24
-rw-rw-r-- 1 simmsben cis90 15 Oct 15 11:13 file1
-rwxr-xr-x 1 simmsben cis90 15 Oct 15 11:13 file2
-r-xr-xr-- 1 simmsben cis90 15 Oct 15 11:13 file3
```

```
[simmsben@opus ~]$ ls Directory1
file1 file2 file3
[simmsben@opus ~]$ cat Directory1 / *
blah blah blah
blah blah blah
blah blah blah
[simmsben@opus ~]$
```



yes he can

Examples group read files

File Permissions

exercise – can **group user** read these files?



```
[simmsben@opus Directory1]$ ls -ld
drwxr-x--x 2 simmsben cis90 4096 Oct 15 12:42 .
[simmsben@opus Directory1]$
[simmsben@opus Directory1]$ ls -l
total 24
-rw-rw-r-- 1 simmsben cis90 15 Oct 15 11:13 file1
-rwxr-xr-x 1 simmsben cis90 15 Oct 15 11:13 file2
-r-xr-xr-- 1 simmsben cis90 15 Oct 15 11:13 file3
```



Can Duke read Benji's files?

File Permissions

exercise – can **group user** read these files?



Can Duke read Benji's files?

```
[simmsben@opus Directory1]$ ls -ld
drwxr-x--x 2 simmsben cis90 4096 Oct 15 12:42 .
[simmsben@opus Directory1]$
[simmsben@opus Directory1]$ ls -l
total 24
-rw-rw-r-- 1 simmsben cis90 15 Oct 15 11:13 file1
-rwxr-xr-x 1 simmsben cis90 15 Oct 15 11:13 file2
-r-xr-xr-- 1 simmsben cis90 15 Oct 15 11:13 file3
```



Yep!

```
[roddyduk@opus ~]$ cat ../simmsben/Directory1/file*
blah blah blah
blah blah blah
blah blah blah
[roddyduk@opus ~]$
```



Examples other read files

File Permissions

exercise – can **other user** read these files?



```
[simmsben@opus ~]$ ls -ld /home /home/cis90 /home/cis90/simmsben
drwxr-xr-x 16 root      root  4096 Jan 27 19:20 /home
drwxr-x--- 33 guest90   cis90 4096 Feb 19 05:49 /home/cis90
drwxr-xr-x 13 simmsben cis90 4096 Mar 30 06:47 /home/cis90/simmsben
```

```
[simmsben@opus Directory1]$ ls -l
total 24
-rw-rw-r-- 1 simmsben cis90 15 Oct 15 11:13 file1
-rwxr-xr-x 1 simmsben cis90 15 Oct 15 11:13 file2
-r-xr-xr-- 1 simmsben cis90 15 Oct 15 11:13 file3
```



Can Mary read Benji's files?

File Permissions

exercise – can **other user** read these files?

Can Mary read Benji's files?



```
[simmsben@opus ~]$ ls -ld /home /home/cis90 /home/cis90/simmsben
drwxr-xr-x 16 root      root  4096 Jan 27 19:20 /home
drwxr-x--- 33 guest90  cis90 4096 Feb 19 05:49 /home/cis90
drwxr-xr-x 13 simmsben cis90 4096 Mar 30 06:47 /home/cis90/simmsben
```

```
[simmsben@opus Directory1]$ ls -l
total 24
-rw-rw-r-- 1 simmsben cis90 15 Oct 15 11:13 file1
-rwxr-xr-x 1 simmsben cis90 15 Oct 15 11:13 file2
-r-xr-xr-- 1 simmsben cis90 15 Oct 15 11:13 file3
```



Nope! - No read permissions to the /home/cis90 directory

```
[simmsmar@opus ~]$ cat /home/cis90/simmsben/Directory1/file1
cat: /home/cis90/simmsben/Directory1/file1: Permission denied
[simmsmar@opus ~]$ cat /home/cis90/simmsben/Directory1/file2
cat: /home/cis90/simmsben/Directory1/file2: Permission denied
[simmsmar@opus ~]$ cat /home/cis90/simmsben/Directory1/file3
cat: /home/cis90/simmsben/Directory1/file3: Permission denied
[simmsmar@opus ~]$
```



Examples owner execute files

File Permissions

exercise – can **owner** execute these files?

```
==> file1 <==  
#!/bin/bash  
echo "blah blah blah"  
exit 0
```

```
==> file2 <==  
#!/bin/bash  
echo "blah blah blah"  
exit 0
```

```
==> file3 <==  
#!/bin/bash  
echo "blah blah blah"  
exit 0
```



Can Benji run his own files?

```
[simmsben@opus Directory1]$ ls -l  
total 32  
-rw-rw-r-- 1 simmsben cis90 41 Oct 15 13:16 file1  
-rwxr-xr-x 1 simmsben cis90 41 Oct 15 13:17 file2  
-r-xr-xr-- 1 simmsben cis90 41 Oct 15 13:18 file3
```

File Permissions

exercise – can **owner** execute these files?

```
==> file1 <==
#!/bin/bash
echo "blah blah blah"
exit 0
```

```
==> file2 <==
#!/bin/bash
echo "blah blah blah"
exit 0
```

```
==> file3 <==
#!/bin/bash
echo "blah blah blah"
exit 0
```



Can Benji run his own files?

```
[simmsben@opus Directory1]$ ls -l
total 32
-rw-rw-r-- 1 simmsben cis90 41 Oct 15 13:16 file1
-rwxr-xr-x 1 simmsben cis90 41 Oct 15 13:17 file2
-r-xr-xr-- 1 simmsben cis90 41 Oct 15 13:18 file3
```

```
[simmsben@opus Directory1]$ ./file1
-bash: ./file1: Permission denied
[simmsben@opus Directory1]$ ./file2
blah blah blah
[simmsben@opus Directory1]$ ./file3
blah blah blah
```



*He can for ones he has
execute permission for*

Examples group execute files

File Permissions

exercise – can **group user** execute these files?



```
[simmsben@opus Directory1]$ ls -ld
drwxr-x--x 2 simmsben cis90 4096 Oct 15 12:42 .
[simmsben@opus Directory1]$
[simmsben@opus Directory1]$ ls -l
total 24
-rw-rw-r-- 1 simmsben cis90 15 Oct 15 11:13 file1
-rwxr-xr-x 1 simmsben cis90 15 Oct 15 11:13 file2
-r-xr-xr-- 1 simmsben cis90 15 Oct 15 11:13 file3
```



Can Duke run Benji's files?

File Permissions

exercise – can **group user** execute these files?



```
[simmsben@opus Directory1]$ ls -ld
drwxr-x--x 2 simmsben cis90 4096 Oct 15 12:42 .
[simmsben@opus Directory1]$
[simmsben@opus Directory1]$ ls -l
total 24
-rw-rw-r-- 1 simmsben cis90 15 Oct 15 11:13 file1
-rwxr-xr-x 1 simmsben cis90 15 Oct 15 11:13 file2
-r-xr-xr-- 1 simmsben cis90 15 Oct 15 11:13 file3
```



Can Duke run Benji's files?

```
[roddyduk@opus ~]$ /home/cis90/simmsben/Directory1/file1
-bash: /home/cis90/simmsben/Directory1/file1: Permission denied
[roddyduk@opus ~]$ /home/cis90/simmsben/Directory1/file2
blah blah blah
[roddyduk@opus ~]$ /home/cis90/simmsben/Directory1/file3
blah blah blah
[roddyduk@opus ~]$
```



He can for the ones he has execute permission for

Examples other execute files

File Permissions

exercise – can **other user** execute these files?



```
[simmsben@opus /]$ ls -ld /home /home/cis90 /home/cis90/simmsben
drwxr-xr-x 16 root      root  4096 Jan 27 19:20 /home
drwxr-x--- 33 rsimms    cis90 4096 Feb 19 05:49 /home/cis90
drwxr-xr-x 13 simmsben cis90 4096 Mar 30 06:47 /home/cis90/simmsben
```

```
[simmsben@opus Directory1]$ ls -l
total 24
-rw-rw-r-- 1 simmsben cis90 15 Oct 15 11:13 file1
-rwxr-xr-x 1 simmsben cis90 15 Oct 15 11:13 file2
-r-xr-xr-- 1 simmsben cis90 15 Oct 15 11:13 file3
```



Can Mary execute Benji's files?

File Permissions

exercise – can **other user** execute these files?



```
[simmsben@opus ~]$ ls -ld /home /home/cis90 /home/cis90/simmsben
drwxr-xr-x 16 root      root  4096 Jan 27 19:20 /home
drwxr-x--- 33 rsimms    cis90 4096 Feb 19 05:49 /home/cis90
drwxr-xr-x 13 simmsben  cis90 4096 Mar 30 06:47 /home/cis90/simmsben
```

```
[simmsben@opus Directory1]$ ls -l
total 24
-rw-rw-r-- 1 simmsben cis90 15 Oct 15 11:13 file1
-rwxr-xr-x 1 simmsben cis90 15 Oct 15 11:13 file2
-r-xr-xr-- 1 simmsben cis90 15 Oct 15 11:13 file3
```

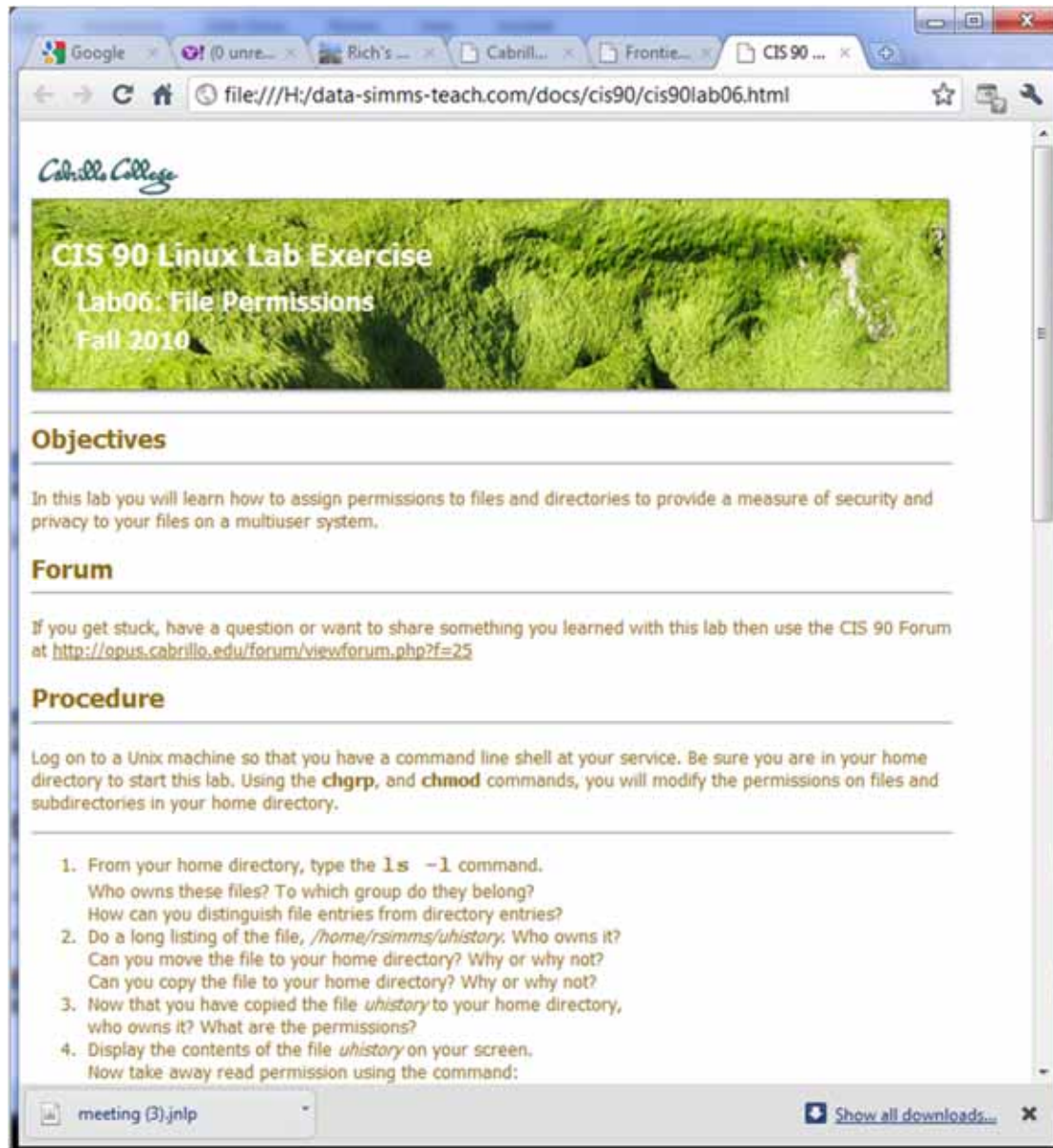


Nope, Mary is locked out at the /home/cis90 level

```
[simmsmar@opus ~]$ /home/cis90/simmsben/Directory1/file1
-bash: /home/cis90/simmsben/Directory1/file1: Permission denied
[simmsmar@opus ~]$ /home/cis90/simmsben/Directory1/file2
-bash: /home/cis90/simmsben/Directory1/file2: Permission denied
[simmsmar@opus ~]$ /home/cis90/simmsben/Directory1/file3
-bash: /home/cis90/simmsben/Directory1/file3: Permission denied
[simmsmar@opus ~]$
```



Lab 6



Google (0 unre... Rich's... Cabrill... Frontie... CIS 90 ...

file:///H:/data-simms-teach.com/docs/cis90/cis90lab06.html

Cabrillo College

CIS 90 Linux Lab Exercise

Lab06: File Permissions

Fall 2010

Objectives

In this lab you will learn how to assign permissions to files and directories to provide a measure of security and privacy to your files on a multiuser system.

Forum

If you get stuck, have a question or want to share something you learned with this lab then use the CIS 90 Forum at <http://opus.cabrillo.edu/forum/viewforum.php?f=25>

Procedure

Log on to a Unix machine so that you have a command line shell at your service. Be sure you are in your home directory to start this lab. Using the **chgrp**, and **chmod** commands, you will modify the permissions on files and subdirectories in your home directory.

1. From your home directory, type the **ls -l** command.
Who owns these files? To which group do they belong?
How can you distinguish file entries from directory entries?
2. Do a long listing of the file, */home/rsimms/uhistory*. Who owns it?
Can you move the file to your home directory? Why or why not?
Can you copy the file to your home directory? Why or why not?
3. Now that you have copied the file *uhistory* to your home directory, who owns it? What are the permissions?
4. Display the contents of the file *uhistory* on your screen.
Now take away read permission using the command:

meeting (3).jnlp Show all downloads...

In this lab you will assign permissions to your files to provide a measure of security

Wrap up

New commands:

chgrp
chmod
chown
groups
stat
umask

change file's group
change file permissions
change file owner (superuser only)
show group membership
show all file inode information
change permission mask

New Files and Directories:

/etc/group

Next Class

Assignment: Check Calendar Page on web site to see what is due next week.

Quiz questions for next class:

Lab 6
Five Posts

- With a umask of 002 what permissions would a newly created file have?
- What is the numeric permission equivalent of `rw-r-xr-x` ?
- Does `chmod o+w` give write permission to the owner or to other users?

Backup

Test 1 – Most Missed Questions

[QX2] Which shell program will be started for millehom (Homer Miller) when he logs into Opus?

```
/home/cis90/simmsben $ cat /etc/passwd
```

```
root:x:0:0:root:/root:/bin/bash
```

```
< snipped >
```

```
ojedavic:x:1252:90:Victor Ojeda:/home/cis90/ojedavic:/bin/bash
```

```
millehom:x:1253:90:Homer Miller:/home/cis90/millehom:/bin/sh
```

```
alvesdes:x:1254:172:Desmond Alves:/home/cis172/alvesdes:/bin/bash
```

```
< snipped >
```

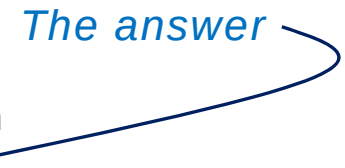
```
mikedel:x:1276:172:Delfino Mike:/home/cis172/mikedel:/bin/bash
```

```
seanbla:x:1277:172:Black Sean:/home/cis172/seanbla:/bin/bash
```

```
veracroc:x:1278:172:Roche Verachai:/home/cis172/veracroc:/bin/bash
```

```
/home/cis90/simmsben $
```

The answer



The last field in /etc/passwd is the shell program to run when the user logs in

/bin/sh is the original shell program that /bin/bash was based on

Test 1 – Most Missed Questions

[Q8] Is the directory `/usr/kerberos/` in your path?

```
/home/cis90/simmsben $ echo $PATH
```

```
/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:/home/cis90/simmsben/./bin:/home/cis90/simmsben/bin:.
```

```
/home/cis90/simmsben $ ls /usr/kerberos/  
bin  man  sbin  share
```

```
/home/cis90/simmsben $ ls /usr/kerberos/bin  
ftp          kinit        krb524init   krsh  rcp      sclient      uuclient  
gss-client    klist        krb5-config  ksu   rlogin   sim_client    v4rcp  
kdestroy     kpasswd      krlogin      kvno  rsh      telnet  
/home/cis90/simmsben $
```

Answer: no

The `/usr/kerberos/bin` directory is in your path, but the `/usr/kerberos` directory is not. These are two different directories.

Test 1 – Most Missed Questions

[Q14] When the command `ls p*[23]` is run from your home directory, what arguments are provided to the `ls` command to process?

Note: It the shell, not the command that parses the command line. The shell parses the command line and expands any files names. After the command line is parsed, the shell locates the command to run on the user's path. The command is then loaded into memory and it runs. The command will get an ordered list of the expanded options and arguments parsed by the shell.

```
/home/cis90/simmsben $ ls p*[23]  
proposal2  proposal3
```

*The **ls** command never sees the "p*[23]". The shell expands p*[23] into two arguments: proposal2 and proposal3. These arguments are then given to the ls command to process.*

Use the echo command as a tool to preview how the shell will do an actual expansion.

```
/home/cis90/simmsben $ echo p*[23]  
proposal2 proposal3
```

Answer

Test 1 – Most Missed Questions

[QX1] In what directory does the dig command reside?

```
/home/cis90/simmsben $ type dig  
dig is /usr/bin/dig
```

The dig command is in the /usr/bin directory

Answer

```
/home/cis90/simmsben $ ls /usr/bin
```

```
[
```

```
411toppm
```

```
< snipped >
```

```
diffstat
```

```
dig
```

```
dir
```

```
< snipped >
```

```
nroff
```

```
nsgmls
```

```
/home/cis90/simmsben $
```

```
nslookup
```

```
nspr-config
```

```
pnmalias
```

```
pnmarith
```

```
pnmcats
```

```
zsoelim
```

Test 1 – Most Missed Questions

[Q2] In what file are all passwords stored?

```
/home/cis90/simmsben $ tail /etc/passwd
poncimar:x:1269:172:Maria Ponciano:/home/cis172/poncimar:/bin/bash
rochever:x:1270:172:Verachai Roche:/home/cis172/rochever:/bin/bash
schreche:x:1271:172:Chelsea Schreiber:/home/cis172/schreche:/bin/bash
schwajoe:x:1272:172:Joel Schwartz:/home/cis172/schwajoe:/bin/bash
tatlojas:x:1273:172:Jason Tatlock:/home/cis172/tatlojas:/bin/bash
velasjos:x:1274:172:Jose Velasco:/home/cis172/velasjos:/bin/bash
lukewat:x:1275:172:Watts Luke:/home/cis172/lukewat:/bin/bash
mikedel:x:1276:172:Delfino Mike:/home/cis172/mikedel:/bin/bash
seanbla:x:1277:172:Black Sean:/home/cis172/seanbla:/bin/bash
veracroc:x:1278:172:Roche Verachai:/home/cis172/veracroc:/bin/bash
```

```
/home/cis90/simmsben $ tail /etc/shadow
tail: cannot open `/etc/shadow' for reading: Permission denied
/home/cis90/simmsben $
```

Answer: /etc/shadow

Note: this was a Flashcards question

Card 8 of 22 (front and back)

In what file are all the account passwords stored?

/etc/shadow

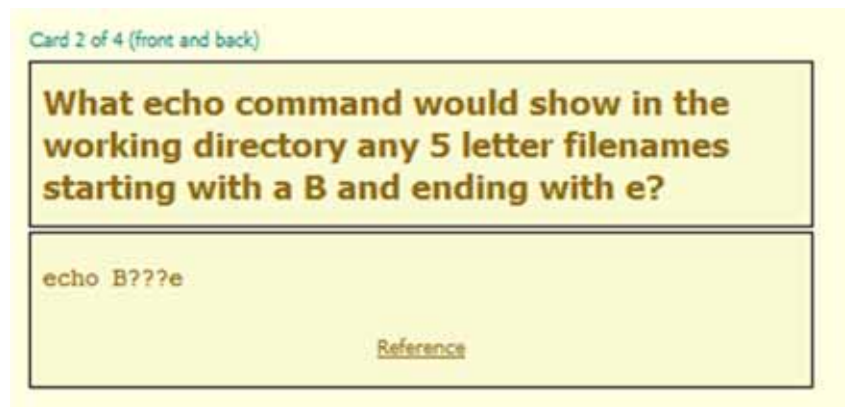
Reference

Test 1 – Most Missed Questions

[Q5] What echo command would show in the working directory any 5 letter filenames starting with a “B” and ending with an “e”?

Answer: echo B???e

Note: this was a Flashcards question



```
/home/cis90/simmsben $ cd Poems/  
/home/cis90/simmsben/Poems $ echo B???e  
Blake
```


File Permissions

Every user has a unique user ID (uid)

```
/home/cis90/simmsben $ id simmsben  
uid=1200(simmsben) gid=90(cis90) groups=90(cis90),100(users)  
context=user_u:system_r:unconfined_t
```

File Permissions

Every user belongs to one or more groups

```
/home/cis90/simmsben $ id simmsben  
uid=1200(simmsben) gid=90(cis90) groups=90(cis90),100(users)  
context=user_u:system_r:unconfined_t
```

File Permissions

User IDs (uid) and primary groups (gid) are stored in /etc/passwd

```
[rsimms@opus test01]$ cat /etc/passwd | grep simmsben  
simmsben:x:1200:90:Benji Simms:/home/cis90/simmsben:/bin/bash  
[rsimms@opus test01]$
```

*Note: We will be learning about the **grep** command in a future lesson*

File Permissions

Relevant fields from the inode

Permissions are shown as triplets using either letters or numbers

- A. Owner** of the file: *uid*
- B. Group** Membership of the file: *gid*
- C. Permissions:** *read, write, execute*

rwX	rwX	rwX
owner	group	other

*These permissions would be
777 using numeric form*

1. the **R**ead permission has a numeric value of 4 (100 binary)
2. the **W**rite permission has a numeric value of 2 (010 binary)
3. the **EX**ecute permission has a numeric value of 1 (001 binary)

The total permissions assigned to a file for a particular identity is the sum of these three values.