

Lesson Module Checklist

- Slides –
- Flash cards –
- Page numbers –
- 1st minute quiz –
- Web Calendar summary –
- Web book pages –
- Commands –

- Lab tested –
- Put uhistory in /home/rsimms/uhistory –

- CCC Confer wall paper / quiz emailed –

- Pick up Polycom phone/extension mics –
- Check that headset is charged –
- Wireless lapel mic backup battery –
- Backup slides, CCC info, handouts on flash drive –



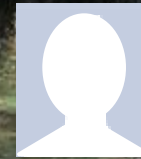
Dieskau



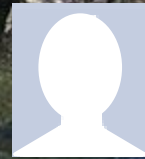
Jonathan



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



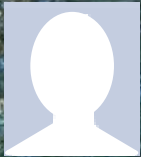
Ana



David



Obie



Dave



Cole



Corey



Nancy



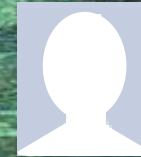
Ryan



Elia



Tasha



Darren



Scott



Devin



Everett



Juan



Raven



Rogan



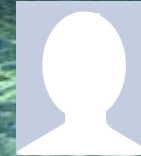
Mike



Mook



Melissa



Cameron



Jose



Jeff



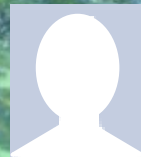
Matt



Kenneth



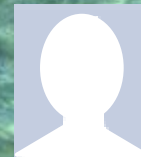
Ousmane



Ian



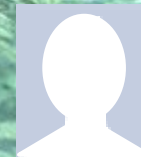
Solomon



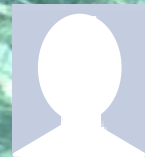
Henry



Matthew



Mason



Chan

First Minute Quiz

Please answer these questions **in the order** shown:

**email answers to: risimms@cabrillo.edu
(within the first few minutes of class)**



- [] Has the phone bridge been added?
- [] Is recording on?
- [] Share slides, putty x 3, Chrome, VLab
- [] Disable spelling on PowerPoint

File Permissions

Objectives

- 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

Agenda

- Quiz
- Review test results
- Question on previous material
- File permissions
- Wrap up

Questions

Previous material and assignment

1. Questions on last lesson?
2. Questions on Lab 5?
3. Questions on Test #1?

Test 1

Post Mortem

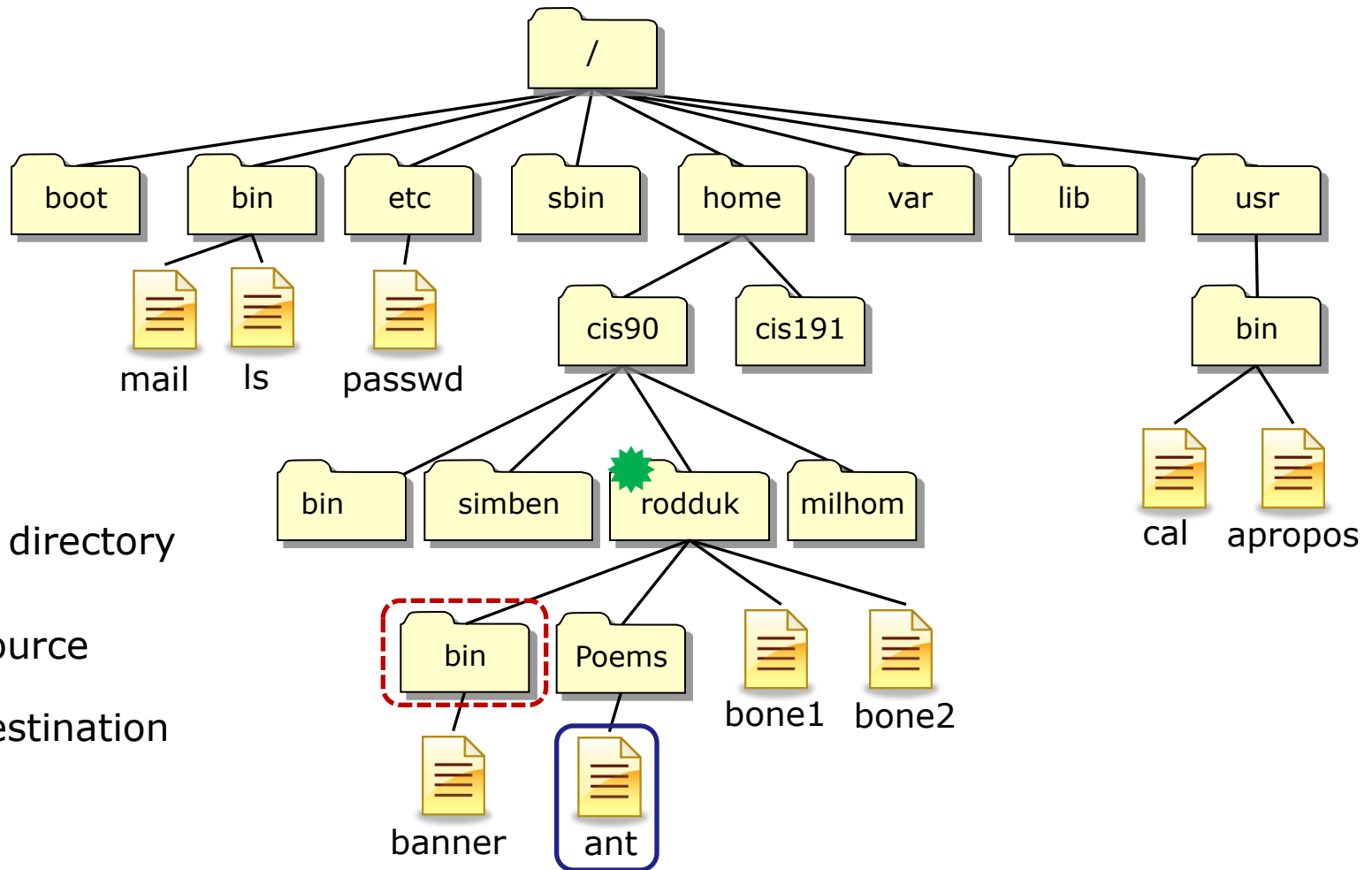
Test 1 – Top and Bottom Ten most missed

Missed Q35 = 23 *recognizing cmds*
 Missed Q33 = 16 *recognizing cmds*
 Missed Q12 = 16 *directory pathnames*
 Missed Q34 = 13 */etc/passwd contents*
 Missed Q22 = 13 *hidden files*
 Missed Q6 = 11 *how shell works*
 Missed Q32 = 11 *prompts*
 Missed Q20 = 11 *relative pathname*
 Missed Q18 = 11 *PATH contents*
 Missed Q26 = 10 *how the shell works*
 Missed Q25 = 9
 Missed Q11 = 9
 Missed Q31 = 8
 Missed Q21 = 7
 Missed Q2 = 6
 Missed Q19 = 6
 Missed Q1 = 6
 Missed Q15 = 6

Missed Q15 = 6
 Missed Q9 = 5
 Missed Q29 = 4
 Missed Q24 = 4
 Missed Q10 = 4
 Missed Q7 = 3
 Missed Q4 = 3
 Missed Q30 = 3
 Missed Q27 = 3 *file types*
 Missed Q16 = 3 *using quotes*
 Missed Q8 = 2 *PATH*
 Missed Q5 = 2 *Linus Torvalds*
 Missed Q23 = 2 *using file command*
 Missed Q17 = 2 *The freebie question*
 Missed Q13 = 2 *using man page*
 Missed Q28 = 1 *using wc command*
 Missed Q3 = 0 *recognizing pathname*
 Missed Q14 = 0 *reading mail*

Review

Pathname Practice



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

```
cp Poems/ant bin/
```

Parse: `cp Poems/ant bin/`

Shell prints a
prompt (using
the `PS1` variable)

Shell parses this command line



The **command** will be loaded only if the shell can locate it on your path (defined by the `PATH` variable)

Options modify the behavior of the command

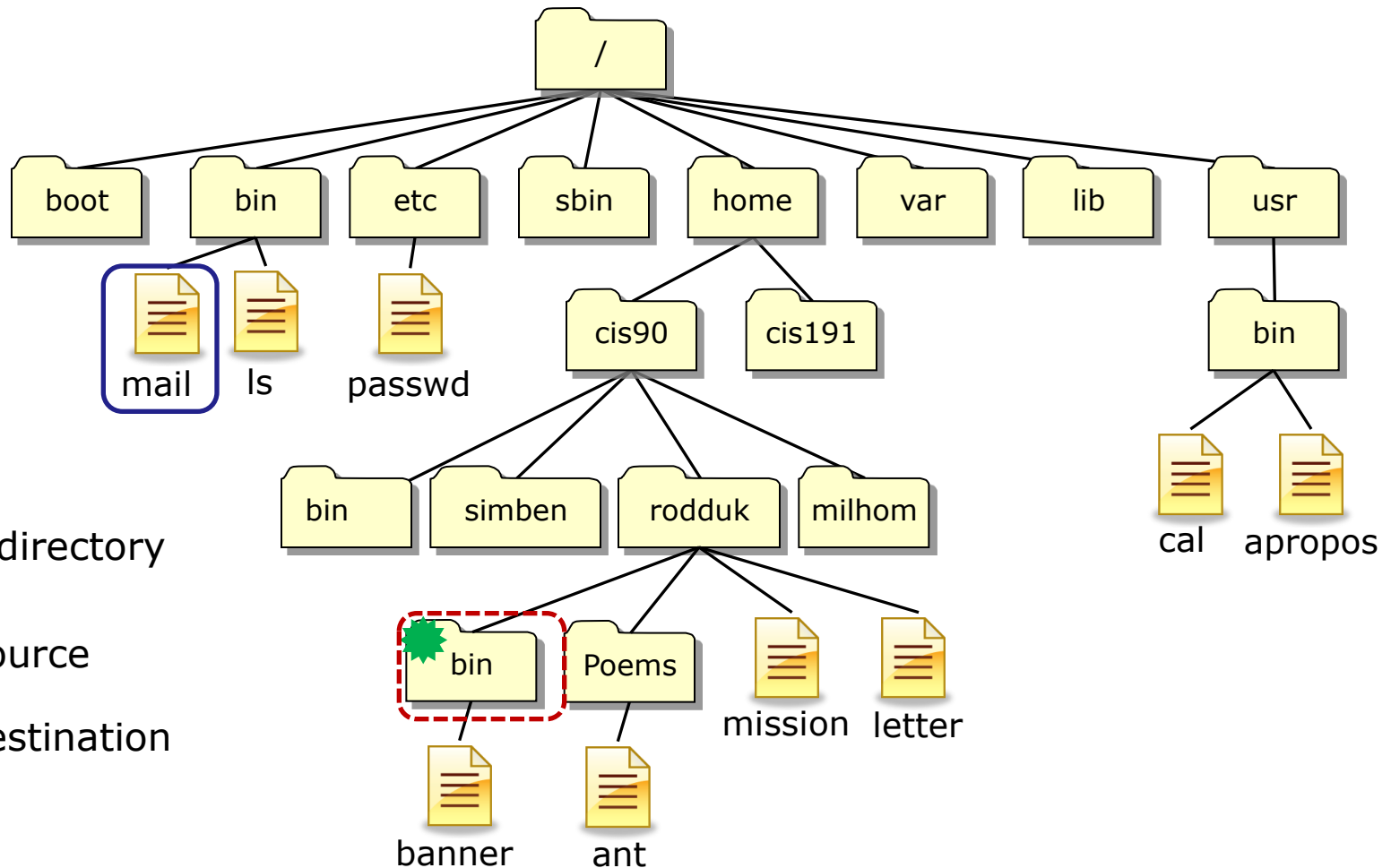
Arguments are what the command works upon

Spaces (blanks) are used to separate the command, options and arguments.

Redirection of `stdin`, `stdout`, `stderr` allows redirecting command input and output

```
/home/cis90/simmsben $ cp  Poems/ant bin/
```

Pathname Practice



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

`cp /bin/mail .`

Parse: **cp /bin/mail .**

Shell prints a
prompt (using
the PS1 variable)

Shell parses this command line



The **command** will be loaded only if the shell can locate it on your path (defined by the PATH variable)

Options modify the behavior of the command

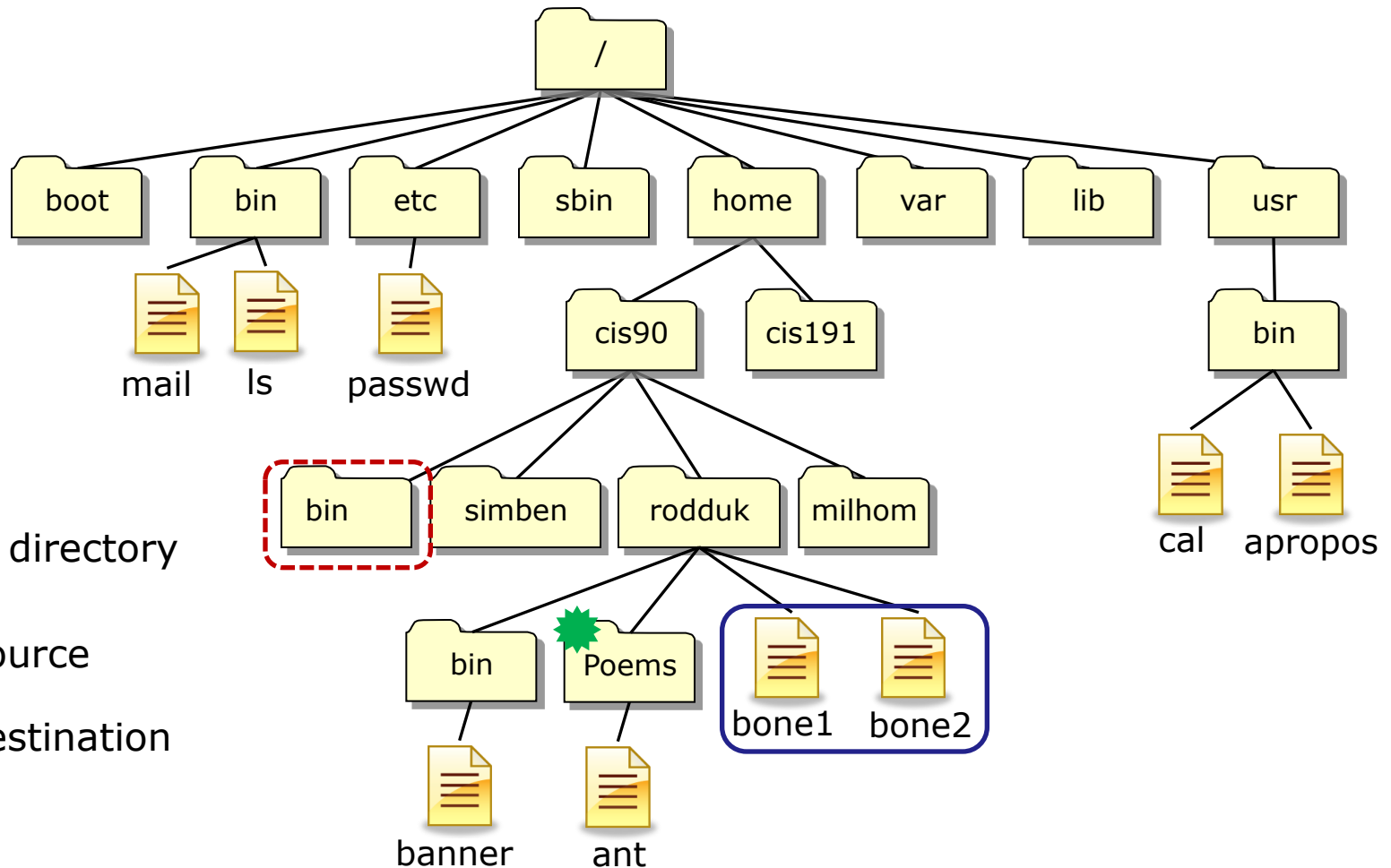
Arguments are what the command works upon

Spaces (blanks) are used to separate the command, options and arguments.

Redirection of stdin, stdout, stderr allows redirecting command input and output

```
/home/cis90/simmsben $ cp /bin/mail .
```

Pathname Practice



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

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

Parse: `mv ../bone? ../../bin/`

Shell prints a
prompt (using
the PS1 variable)

Shell parses this command line



The **command** will be loaded only if the shell can locate it on your path (defined by the PATH variable)

Options modify the behavior of the command

Arguments are what the command works upon

Spaces (blanks) are used to separate the command, options and arguments.

Redirection of stdin, stdout, stderr allows redirecting command input and output

/home/cis90/simmsben \$ `mv` `../bone?` `../../bin/`

The shell, not the command, processes any filename expansion metacharacters

/home/cis90/simmsben \$ `mv` `../bone1` `../bone2` `../../bin/`



Housekeeping

Housekeeping



Rich's Cabrillo College CIS Classes

CIS 90 Grades

[Home](#)
[Resources](#)
[Forums](#)
[CIS Lab](#)
[CTC](#)

[Login](#)
[Flashcards](#)
[Admin](#)
[CIS 90](#)
[Previous Classes](#)
[75 days till term ends!](#)
[Cabrillo College](#)
[Web Advisor](#)
[Static IDs](#)
[Commands and Files](#)
[Accessing VLab](#)
[RIP Dennis Ritchie](#)

CIS 90 (Spring 2012) Grades

[Course Home](#)
[Calendar](#)

Points can be earned from the following activities:

- 5% - Quizzes
- 16% - Tests
- 14% - Help forum participation
- 54% - Lab assignments
- 11% - Final project

How your grade is determined:
A student can earn up to 560 total points doing the activities listed above. The course grade is based on the number of points earned.

Percentage	Total Points	Letter Grade	Pass/No Pass
90% or higher	504 or higher	A	Pass
80% to 89.9%	448 to 503	B	Pass
70% to 79.9%	392 to 447	C	Pass
60% to 69.9%	336 to 391	D	No pass
0% to 59.9%	0 to 335	F	No pass

For some flexibility, personal preferences or family emergencies there is an additional 90 points available of **extra credit** activities.

Choice of Grade or Pass/No Pass
You indicate your grading choice on the Student Survey form passed out during the first class. You can verify your grading choice selection on the table below. Contact the instructor by email with any questions or to request a change in grading choice.

Recommendations
The instructor may provide letters of recommendation upon request. When writing a recommendation the instructor will include both graded and non-graded areas of performance. Non-graded performance areas may include teamwork, helping others, quality, planning & organization skills, communication, documentation, motivation, and the desire to go above and beyond expectations. The forum is an excellent way to demonstrate teamwork and communication skills.

Current Progress

Code Name	Grading Choice	Quizzes & Tests										Forum										Labs										Project	Extra Credit	Total	Grade
		Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	T1	T2	T3	F1	F2	F3	F4	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10							
Max Points		3	3	3	3	3	3	3	3	3	30	30	30	20	20	20	20	20	30	30	30	30	30	30	30	30	30	30	30	60	90	560			
addenda 10NP		3	1	3	2						28	20							24	28	30	27									18				
alatar Grade	2	3	3	3							28	20							27	30	30	26									23				
amroth Grade	3	3	3	3							28	20							29	23	30	26									20				
arador Grade	3	3	3	3							25	20							19	30	24									24					
aragorn Grade	3	3	3	3							24	20							28	15	28	26									21				
arwen Grade	3	1	3	3							27	20							29	30	30	29									22				
carc Grade	3	3	3	3							24	16							30	29	30	19									21				
celebrian Grade	3	3	3	3							23	20							1	30	29									21					
cedan Grade	3	3	3	3							29	20							28	28	30	24									23				
denethor Grade	2										20								14	22	28									7					
dori Grade	3	2	3	2							23	16							28	27	30									11					
dvain Grade	3	1	1	1							19	20							28	25	28	22									21				
elrond Grade	3	1	3	3							28	20							30	30	28	29									25				
eomer Grade	2	2	1	2							18	20							21	25	30	23									8				
ewyn Grade											26	20							26	28	30	15									22				
frodo Grade	3	3	3	3							26	20							28	24	30	29									25				
gvalah Grade	1	3	2	3							22	20							25	22	30	18									13				
haldir Grade				0							26	20							30	28	28	20									23				
loreth Grade	3	3									12	8							21	30										5					
khamul Grade	3	2	3	2							16	20							29	22	30	22									15				
magul Grade												4							8											2					
nessa Grade												20							30	24	28	20									13				
orome Grade											27	20							30	27	30	25									24				
pippen Grade	3	3	3	2							27	20							30	30	30									15					
quickbeam Grade	3	3	3	3							27	20							35	29	30	28									21				
samwise Grade	2	3	3								23	4							22	16	24									13					
shadowfax Grade	3	3	3								27	20							29	30	30	28									26				
strider Grade	2			3							28	20							30	28	30	29									27				
theoden Grade		0	3								23	0							27	22										4					
treebeard Grade	3	3	3	3							28	16							28	25	30	29									22				
tulkas Grade	3	2		0							22	0							30	22	28	18									4				
ulmo Grade	3	3	3	3							27	20							28	25	30	30									18				
vanda Grade												0																							

[Metal](#)
[Sitemap](#)
[WSC XHTML 1.0](#)
[WSC CSS](#)
[Credits](#)
[Earth](#)

Please monitor your grades on the Grades web page.

If you feel you are not where you want to be then contact me to arrange some extra help.

If you were not able to take Test 1 you may take the makeup version (different questions)

Permissions

R=Read

W=Write

X=Execute



File Permissions Read



Read permission is necessary to read a file

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 Write



Write permission is necessary to write to a file

Benji had write permission to his own letter file

```
/home/cis90/simben $ echo "Benji was here" >> letter
/home/cis90/simben $ tail -4 letter
Mother, Father, kindly disregard this letter.
```

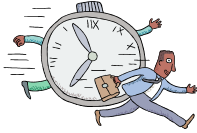
This is how you append a line to a file. We will learn more about >> later!

Alan Sherman

```
Benji was here
```

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

```
/home/cis90/simben $ echo "Benji was here" >> ../rodduk/letter
-bash: ../rodduk/letter: Permission denied
```



File Permissions Execute



Execute permission is necessary to execute (run) a file (command, program or script)

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

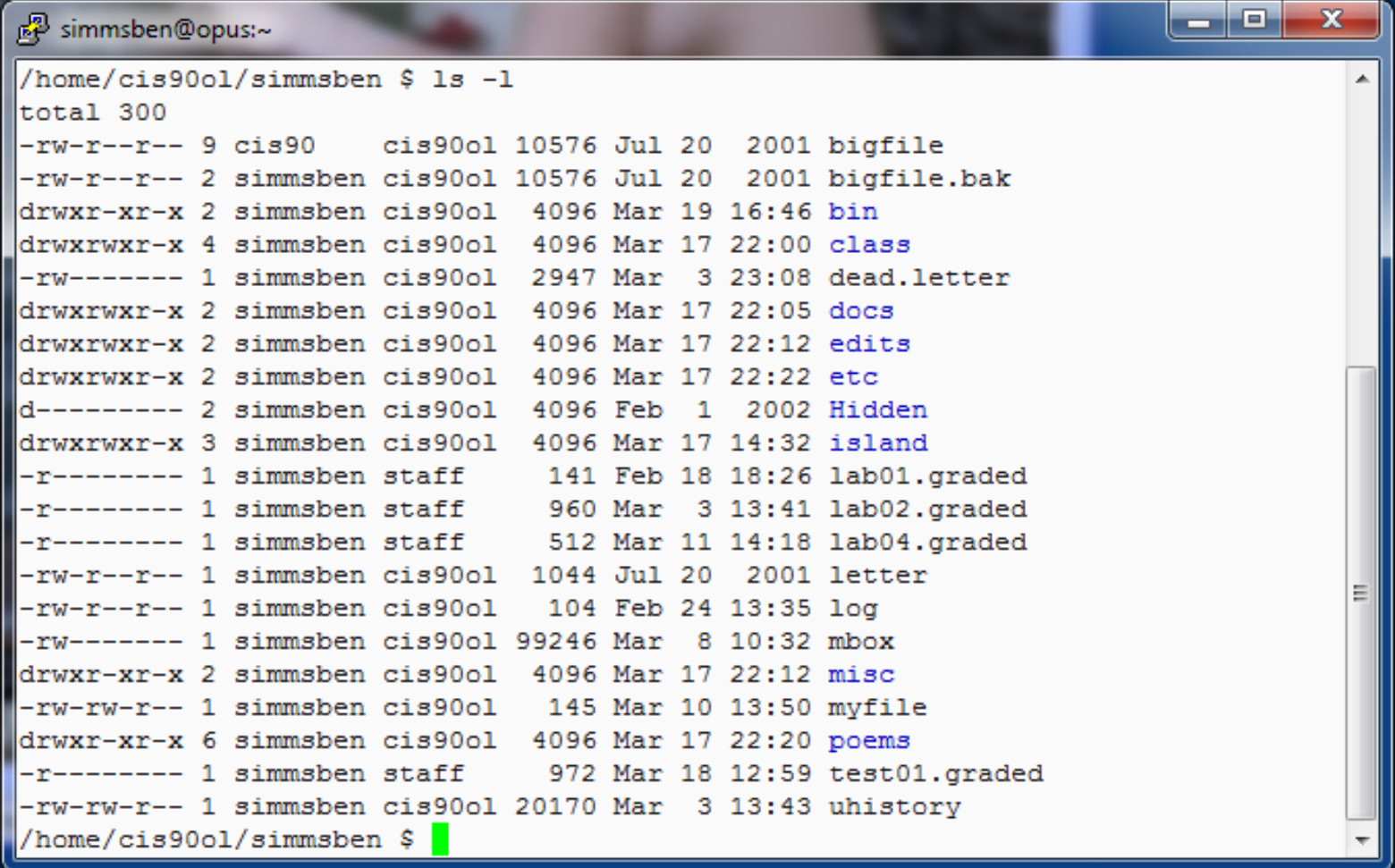
```
/home/cis90/simben $ tryme  
My name is "tryme"  
I am pleased to make your acquaintance, Benji Simms  
/tmp
```

But he does not have execute permission on the check7 script!

```
/home/cis90/simben $ check7  
-bash: /home/cis90/simben/../../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 information

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 $
```

Columns 2-10 shows the permissions on each file using r,w,x or -

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 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-- 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 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 $
```

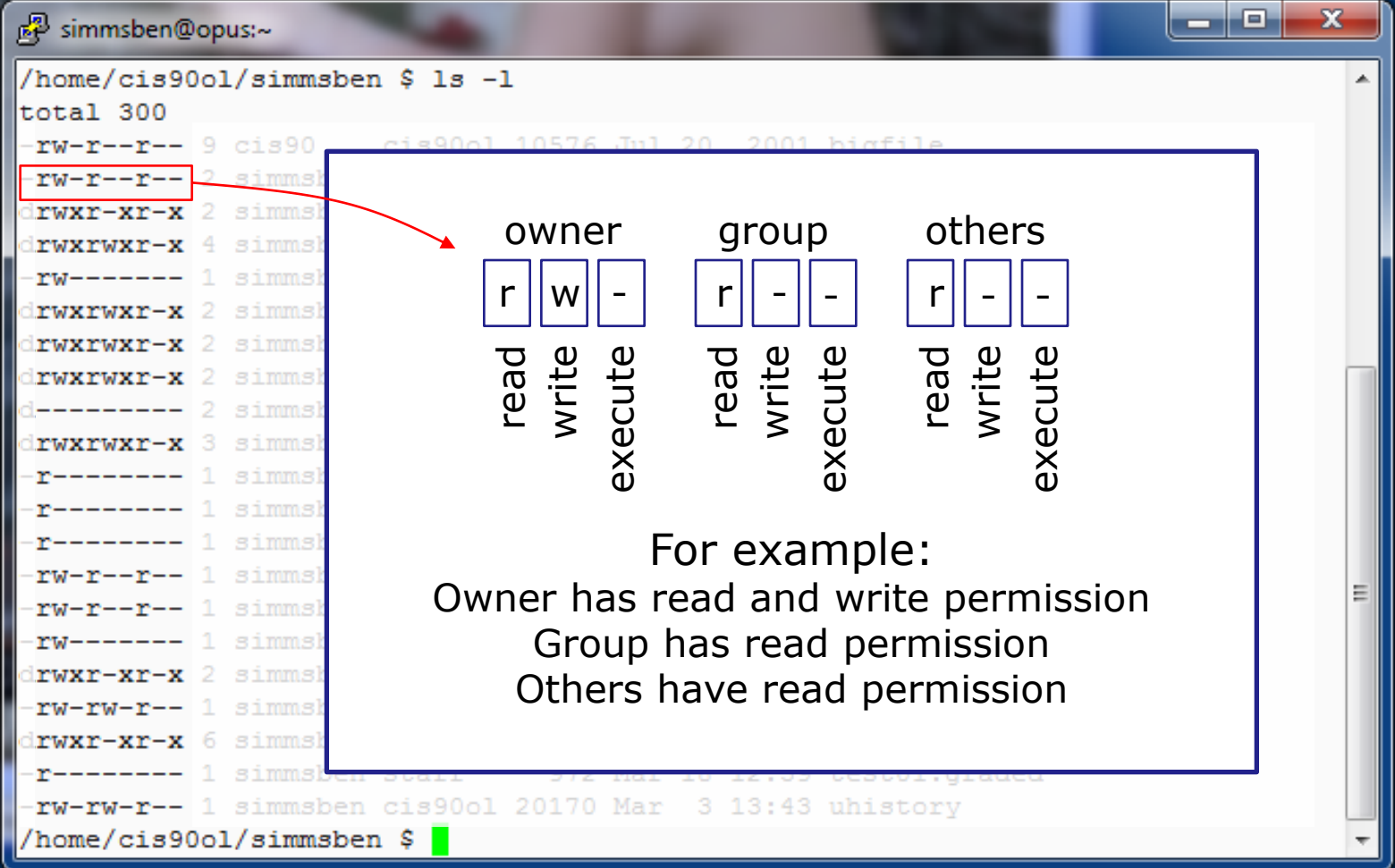
owner			group			others		
read	write	execute	read	write	execute	read	write	execute

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` and its output. A red box highlights the permissions `-rw-r--r--` for the file `bigfile`. A red arrow points from this box to a diagram that breaks down the permissions for owner, group, and others.

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

Permissions may be set for the owner, the group and all others

File Permissions

Group Membership

What groups does Benji and Duke belong to?

```
/home/cis90/simben $ id  
uid=1000(simben90) gid=90(cis90) groups=90(cis90),100(users)  
context=user_u:system_r:unconfined_t
```

```
/home/cis90/simben $ groups  
cis90 users
```

```
/home/cis90/simben $ id rodduk90  
uid=1046(rodduk90) gid=90(cis90) groups=90(cis90),100(users)
```

```
/home/cis90/simben $ groups rodduk90  
rodduk90 : cis90 users
```

*The **id** or **groups** command can show group membership. You may optionally specify a user other than yourself as an argument.*



File Permissions Read



Read permission is necessary to read a file

Benji, considered as "other", 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, as "other", 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/cis90/simben $ ls -l /etc/passwd /etc/shadow
-rw-r--r-- 1 root root 8686 Mar 14 14:08 /etc/passwd
-r----- 1 root root 13045 Mar 20 09:03 /etc/shadow
```

Both these files are owned by root and are in the root group



File Permissions Write



Write permission is necessary to write to a file

Benji, as "owner", has write permission to his own letter file

```
/home/cis90/simben $ echo "Benji was here" >> letter
/home/cis90/simben $ tail -4 letter
Mother, Father, kindly disregard this letter.
```

Alan Sherman

Benji was here

But as member of group cis90 , does not have write permission to Duke's letter file!

```
/home/cis90/simben $ echo "Benji was here" >> ../rodduk/letter
-bash: ../rodduk/letter: Permission denied
```

```
/home/cis90/simben $ ls -l letter ../rodduk/letter
-rw-r--r-- 1 simben90 cis90 1059 Mar 20 10:31 letter
-rw-r--r-- 1 rodduk90 cis90 1044 Jul 20 2001 ../rodduk/letter
```

These files have different owners but are in the same group



File Permissions Execute



Execute permission is necessary to execute (run) a file (command, program or script)

Benji, as "owner", has execute permission on his tryme script

```
/home/cis90/simben $ tryme
My name is "tryme"
I am pleased to make your acquaintance, Benji Simms
/tmp
```

But as "other", he does not have execute permission on check7

```
/home/cis90/simben $ check7
-bash: /home/cis90/simben/../../bin/check7: Permission denied
```

```
/home/cis90/simben $ type tryme check7      find where commands reside on path
tryme is hashed (/home/cis90/simben/bin/tryme)
check7 is hashed (/home/cis90/simben/../../bin/check7)
/home/cis90/simben $ ls -l bin/tryme ../../bin/check7      view permissions
-rwx----- 1 rsimms  staff 8657 Jan 23 14:56 ../../bin/check7
-rwxr-xr-x 1 simben90 cis90 174 Mar  4 2004 bin/tryme
```

Groups and new files

Groups

```
/home/cis90/roddyduk $ touch mydogs  
/home/cis90/roddyduk $ ls -l mydogs  
-rw-rw-r-- 1 roddyduk cis90 0 Oct 19 13:16 mydogs
```

When a new file is created:

- *the owner is set to the user creating the file*
- *the group is set to the user's primary group*

Groups

*Use either **id** or **groups** command to determine what groups a user belongs to*

```
[rsimms@opus lab06]$ id roddyduk  
uid=1201(roddyduk) gid=90(cis90) groups=90(cis90),100(users)  
context=user_u:system_r:unconfined_t  
[rsimms@opus lab06]$
```

```
[roddyduk@opus ~]$ groups roddyduk  
roddyduk : cis90 users
```

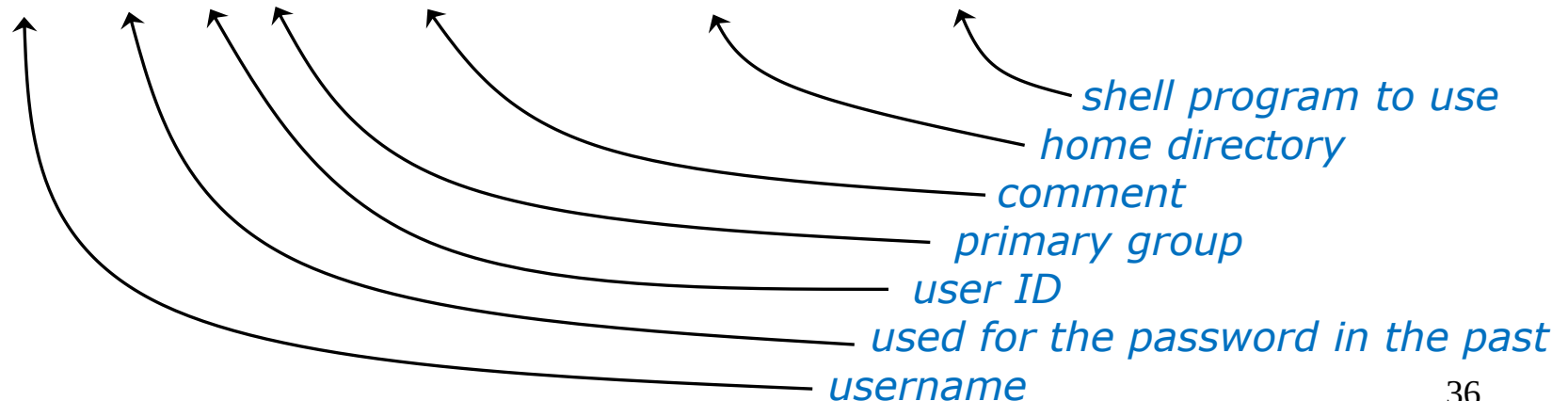
*Primary group (gid) is cis90,
secondary group is users*

Groups

The user's primary group is stored in /etc/passwd (the 4th field)

Excerpt from **/etc/passwd**

```
simmsben:x:1200:90:Benji Simms:/home/cis90/simmsben:/bin/bash
roddyduk:x:1201:90:Duke Roddy:/home/cis90/roddyduk:/bin/bash
clastmax:x:1009:191:Max Clastor:/home/cis191/clastmax:/bin/bash
derriale:x:1202:90:Alex Derrick:/home/cis90/derriale:/bin/bash
garciton:x:1203:90:Tony Garcia:/home/cis90/garciton:/bin/bash
garibjam:x:1204:90:James Garibay:/home/cis90/garibjam:/bin/bash
rochajua:x:1205:90:Juan Rocha:/home/cis90/rochajua:/bin/bash
delfimik:x:1206:90:Mike Delfin:/home/cis90/delfimik:/bin/bash
dingechr:x:1207:90:Christine Dinges:/home/cis90/dingechr:/bin/bash
blacksea:x:1208:90:Sean Black:/home/cis90/blacksea:/bin/bash
antiden:x:1209:90:Dennis Anti:/home/cis90/antiden:/bin/bash
```



Groups

Secondary groups are recorded in /etc/group

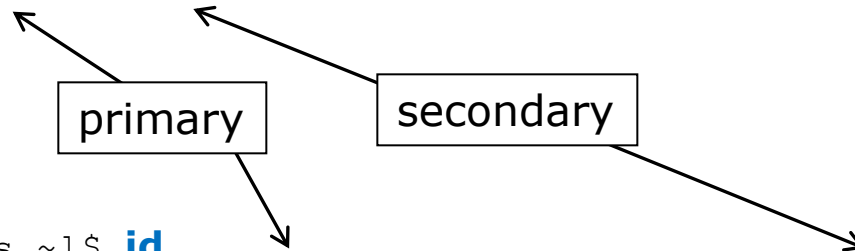
Excerpts from **/etc/group**

```
users:x:100:guest,guest90,jimg,abbenste,arltjef,bolasale,bowerjak,dycktim,farreeli,ga
virxim,gilart,gonzaian,goodmthe,hammoste,kotilnat,lenzpat,maganfra,mattimar,mccarmic,
mchalgeo,mezalui,ortegcar,rochaleo,spadymat,starkmic,vasqucar,vistigab,wallgle,watsoh
ar,quintjos,swansgre,archiand,moonecar,orourpat,pantogab,velasoli,simmsben,roddyduk,c
lastmax,derriale,garciton,garibjam,rochajua,delfimik,dingechr,blacksea,antiden,pirkll
au,birmijam,messison,zilissau,plastadr,brownliz,husemat,botoschr,perezrud,palmilar,sa
linjac,hamiljas,pennitan,valadand,woodjan,henrydal,galbrnat,dakkaabd,cardefra,daviesa
r,hrdinste,redmanic,enriqste,dawadast,menafer,orozcmig,srecklau,mottste,fouric,wattsl
uk,dahlicas,velasliv,pitzemik,komicser,parrijen,beltredt,hernaaar,brownbri,castrsal,m
artiant,joossam,ojedavic,millehom,alvesdes,bejarjoh,bergejoh,breitrob,clarkgal,desotm
at,gardnnic,huangyan,leetheri,lewisgre,studetes,lighttho,lindadav,madrista,normasea,p
oncimar,rochever,schreche,schwajoe,tatlojas,velasjos,lukewat,mikedel,seanbla,veracroc
,simmsmar
utmp:x:22:
utempter:x:35:
< snipped >
mikki:x:501:
guest:x:506:
staff:x:503:jimg,rsimms,gerlinde
cis90:x:90:jimg,guest,rsimms
cis130:x:130:jimg,rsimms
```

Groups

Every user is a member of a **primary group** (shown in /etc/passwd) and multiple **secondary groups** (shown in /etc/group)

```
/home/cis90/roddyduk $ groups roddyduk  
roddyduk : cis90 users
```



```
[roddyduk@opus ~]$ id  
uid=1201(roddyduk) gid=90(cis90) groups=90(cis90),100(users)  
context=user_u:system_r:unconfined_t
```

```
/home/cis90/roddyduk $ touch mydogs  
/home/cis90/roddyduk $ ls -l mydogs  
-rw-rw-r-- 1 roddyduk cis90 0 Oct 19 13:16 mydogs
```

Note, new files are created using the primary group

Specifying Numerical Permissions

File Permissions

Binary

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

rwX	Binary	Convert	Decimal
_ _ _	0 0 0	0 + 0 + 0	0
_ _ x	0 0 1	0 + 0 + 1	1
_ w _	0 1 0	0 + 2 + 0	2
_ w x	0 1 1	0 + 2 + 1	3
r _ _	1 0 0	4 + 0 + 0	4
r _ x	1 0 1	4 + 0 + 1	5
r w _	1 1 0	4 + 2 + 0	6
r w x	1 1 1	4 + 2 + 1	7

4's column ———→
 2's column ———→
 1's column ———→

File Permissions

Binary

rwX	Binary	Convert	Decimal
— — —	0 0 0	0 + 0 + 0	0
— — x	0 0 1	0 + 0 + 1	1
— w —	0 1 0	0 + 2 + 0	2
— w x	0 1 1	0 + 2 + 1	3
r — —	1 0 0	4 + 0 + 0	4
r — x	1 0 1	4 + 0 + 1	5
r w —	1 1 0	4 + 2 + 0	6
r w x	1 1 1	4 + 2 + 1	7

Example: **rw-** (read, write, no execute)

$$\begin{array}{ccccc}
 = 110 & \text{or} & 4+2+0 & = & 6 \\
 \text{binary} & & \text{decimal} & & \text{decimal}
 \end{array}$$

File Permissions

Binary

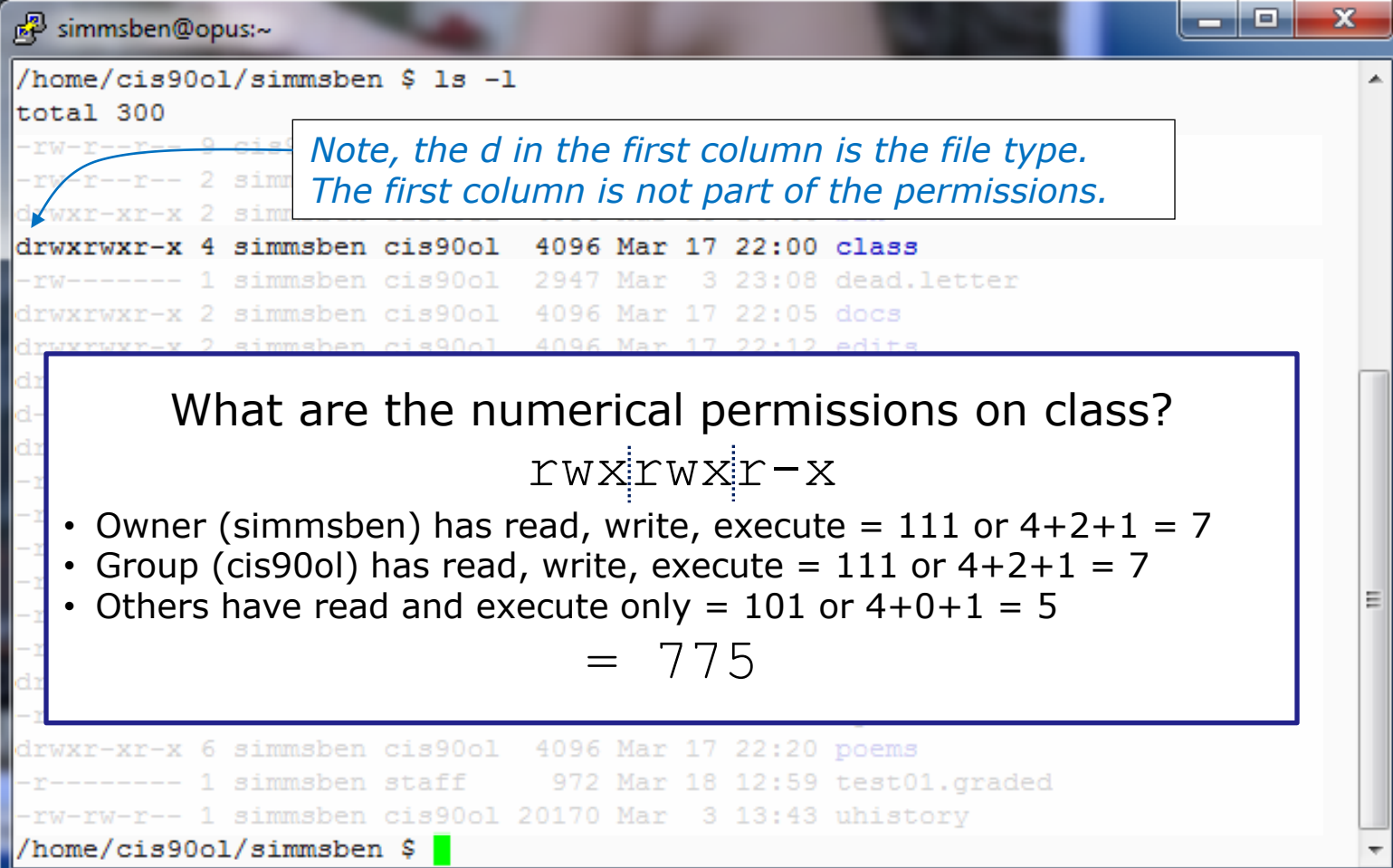
rwX	Binary	Convert	Decimal
— — —	0 0 0	0 + 0 + 0	0
— — x	0 0 1	0 + 0 + 1	1
— w —	0 1 0	0 + 2 + 0	2
— w x	0 1 1	0 + 2 + 1	3
r — —	1 0 0	4 + 0 + 0	4
r — x	1 0 1	4 + 0 + 1	5
r w —	1 1 0	4 + 2 + 0	6
r w x	1 1 1	4 + 2 + 1	7

Example: **-wx** (no read, write, execute)

$$\begin{array}{ccccc}
 = 011 & \text{or} & 0+2+1 & = & 3 \\
 \text{binary} & & \text{decimal} & & \text{decimal}
 \end{array}$$

File Permissions

Let's practice interpreting permissions



The screenshot shows a terminal window with the command `ls -l` executed in the directory `/home/cis90ol/simmsben`. The output lists several files, including `class`, `dead.letter`, `docs`, `editors`, `poems`, `test01.graded`, and `uhistory`. A callout box points to the first column of the permissions string for `class` (`drwxrwxr-x`), stating: "Note, the d in the first column is the file type. The first column is not part of the permissions."

What are the numerical permissions on class?

`rwX|rwX|r-X`

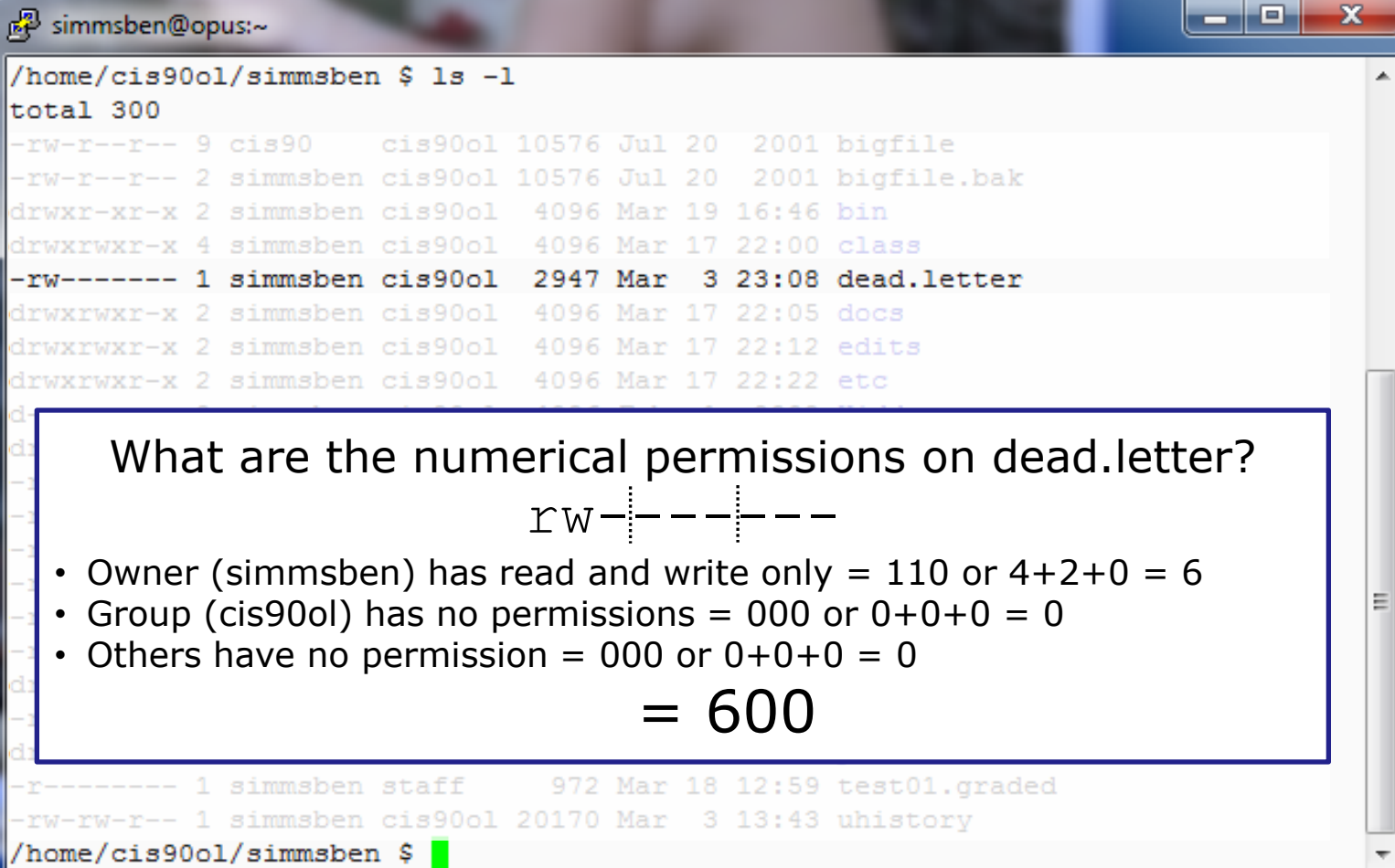
- Owner (simmsben) has read, write, execute = 111 or $4+2+1 = 7$
- Group (cis90ol) has read, write, execute = 111 or $4+2+1 = 7$
- Others have read and execute only = 101 or $4+0+1 = 5$

$= 775$

/home/cis90ol/simmsben/class (directory)

File Permissions

Let's practice interpreting permissions



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
drwxrwxr-x 2 simmsben cis90ol 4096 Mar 17 22:22 misc
drwxrwxr-x 2 simmsben cis90ol 4096 Mar 17 22:22 test01
drwxrwxr-x 2 simmsben cis90ol 4096 Mar 17 22:22 uhistory
-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 $
```

What are the numerical permissions on dead.letter?

$rw- - - - -$

- Owner (simmsben) has read and write only = 110 or $4+2+0 = 6$
- Group (cis90ol) has no permissions = 000 or $0+0+0 = 0$
- Others have no permission = 000 or $0+0+0 = 0$

= 600

/home/cis90ol/simmsben/dead.letter (regular file)

File Permissions

Let's practice interpreting permissions

```
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 edita  
drwx  
d---  
drwx  
-r--  
-r--  
-r--  
-rw-  
-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 13:43 uhistory  
/home/cis90ol/simmsben $
```

What are the numerical permissions on test01.graded?

r--|---|---

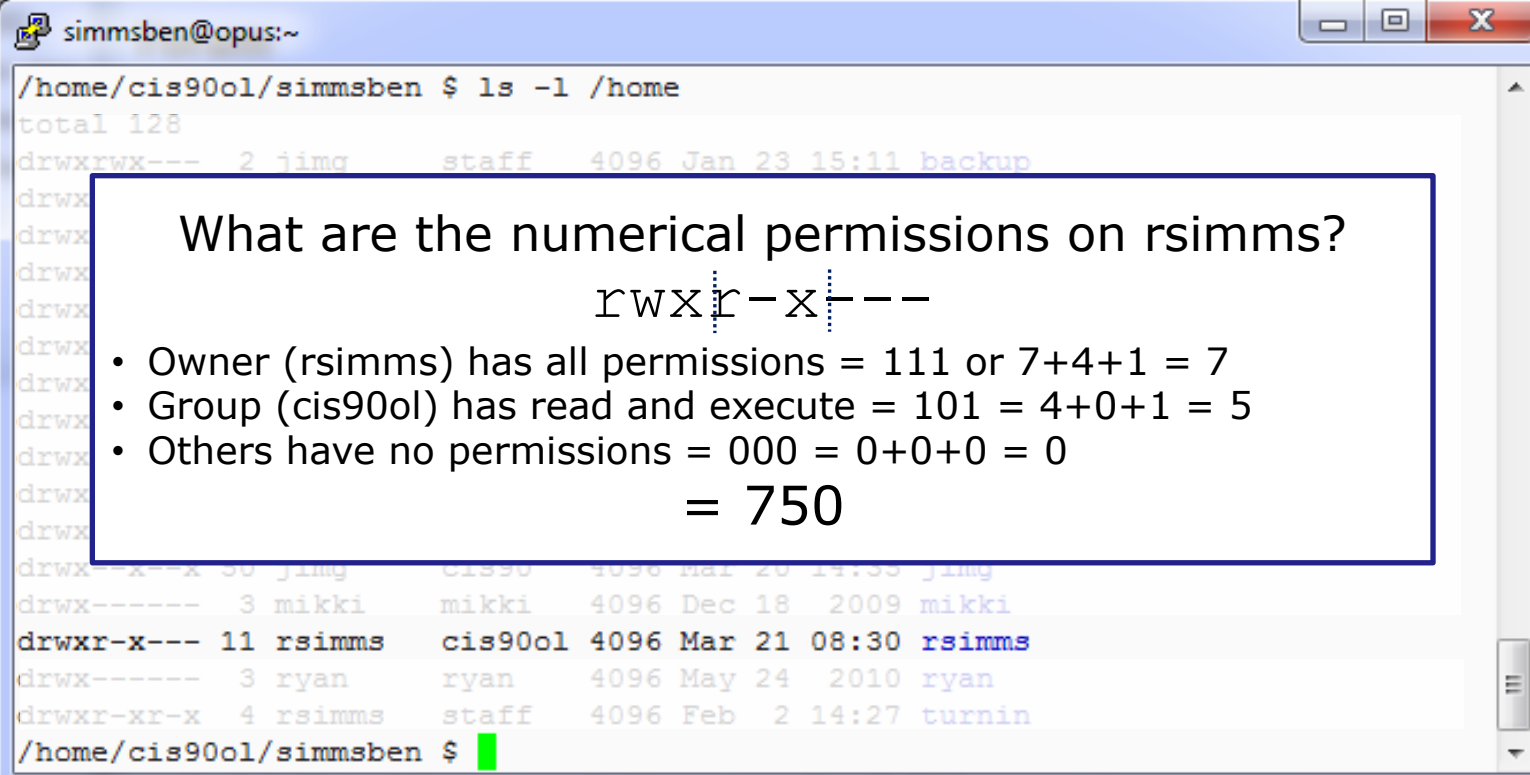
- Owner (simmsben) has read only = 100 or 4+0+0 = 4
- Group (cis90ol) has no permissions = 000 or 0+0+0 = 0
- Others have no permission = 000 or 0+0+0 = 0

= 400

/home/cis90ol/simmsben/test01.graded (regular file)

File Permissions

Let's practice interpreting permissions



The image shows a terminal window with a blue title bar containing the text "simmsben@opus:~". The terminal displays the command `ls -l /home` and its output, which lists several files and directories. A blue-bordered text box is overlaid on the terminal, containing the question "What are the numerical permissions on rsimms?" and an explanation of the permissions `rwXr-x---` for the file `rsimms`. The explanation states that the owner (rsimms) has all permissions (111 or 7+4+1 = 7), the group (cis90ol) has read and execute permissions (101 = 4+0+1 = 5), and others have no permissions (000 = 0+0+0 = 0), resulting in the numerical permission `750`.

```
simmsben@opus:~$ ls -l /home
total 128
drwxrwx--- 2 jimg      staff   4096 Jan 23 15:11 backup
drwxrwx--- 3 jimg      staff   4096 Jan 23 15:11 jimg
drwxrwx--- 3 jimg      staff   4096 Jan 23 15:11 mikki
drwxrwx--- 3 jimg      staff   4096 Jan 23 15:11 ryan
drwxrwx--- 3 jimg      staff   4096 Jan 23 15:11 turnin
drwxrwx--- 3 jimg      staff   4096 Jan 23 15:11 rsimms
drwxrwx--- 3 jimg      staff   4096 Jan 23 15:11 cis90ol
drwxr-x-x 30 jimg      cis90ol 4096 Mar 20 14:33 jimg
drwx----- 3 mikki     mikki   4096 Dec 18 2009 mikki
drwxr-x--- 11 rsimms    cis90ol 4096 Mar 21 08:30 rsimms
drwx----- 3 ryan      ryan    4096 May 24 2010 ryan
drwxr-xr-x 4 rsimms     staff   4096 Feb  2 14:27 turnin
/home/cis90ol/simmsben $
```

What are the numerical permissions on rsimms?

`rwXr-x---`

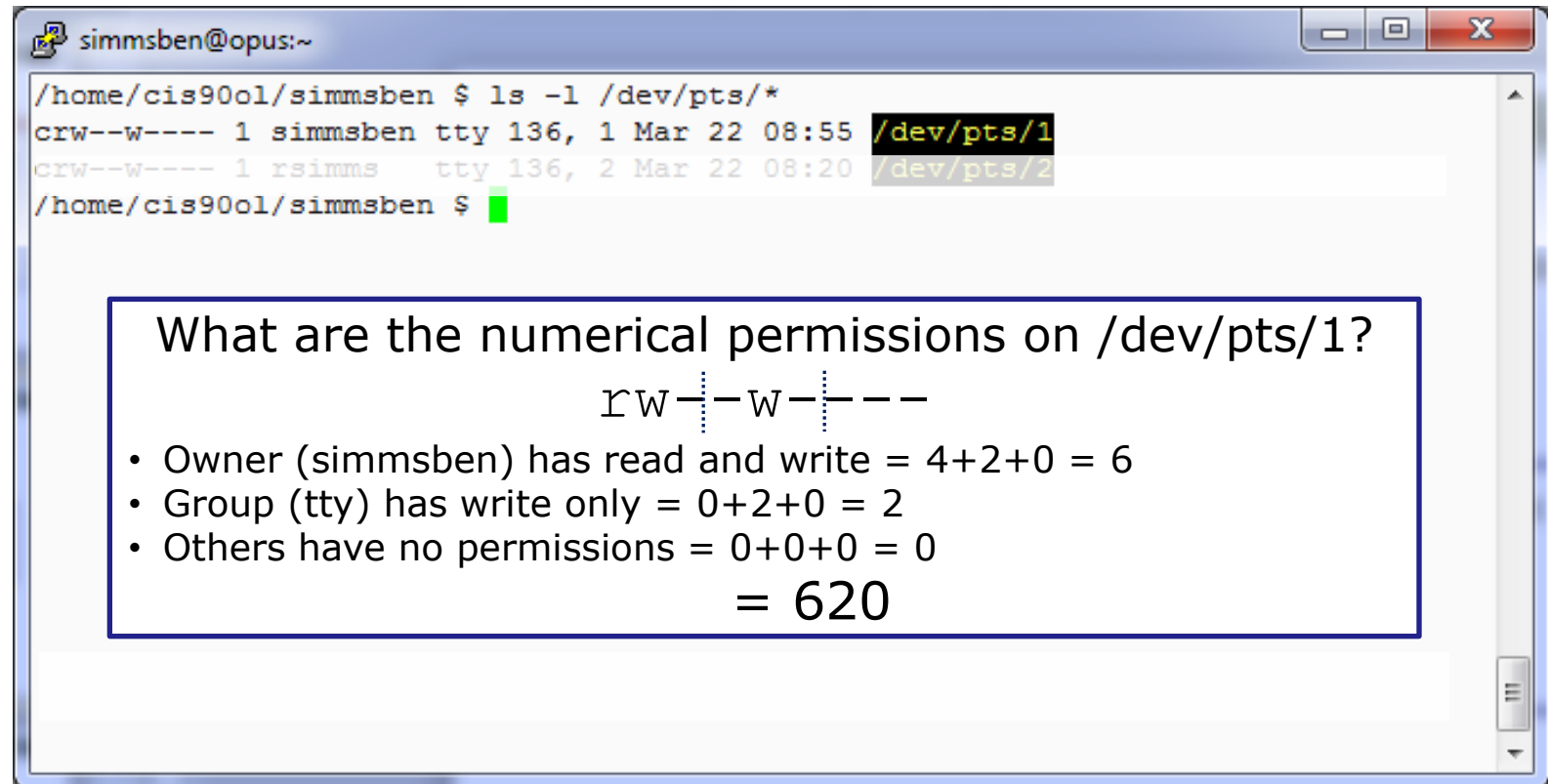
- Owner (rsimms) has all permissions = 111 or 7+4+1 = 7
- Group (cis90ol) has read and execute = 101 = 4+0+1 = 5
- Others have no permissions = 000 = 0+0+0 = 0

= 750

/home/rsimms (directory)

File Permissions

Interpreting the permission codes on long listings



```
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 rsimms tty 136, 2 Mar 22 08:20 /dev/pts/2  
/home/cis90ol/simmsben $
```

What are the numerical permissions on /dev/pts/1?

$\begin{array}{ccccccc} r & w & - & - & w & - & - \\ & | & | & & | & | & \\ & 4 & 2 & & 0 & 2 & 0 \end{array}$

- Owner (simmsben) has read and write = $4+2+0 = 6$
- Group (tty) has write only = $0+2+0 = 2$
- Others have no permissions = $0+0+0 = 0$

= 620

/dev/pts/1 (character special device file)

Recap

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 **user (owner)** 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
lynbeerix: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 >
```

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

```
/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 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 cis90ol groups

```
/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 $ groups simmsben
simmsben : cis90ol users
```

The simmsben user is a member of these groups



```
/home/cis90ol/simmsben $ groups rsimms
rsimms : staff cis90 cis90ol 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,jimg
```

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

bigfile 102574
bin 102575
letter 102609

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.

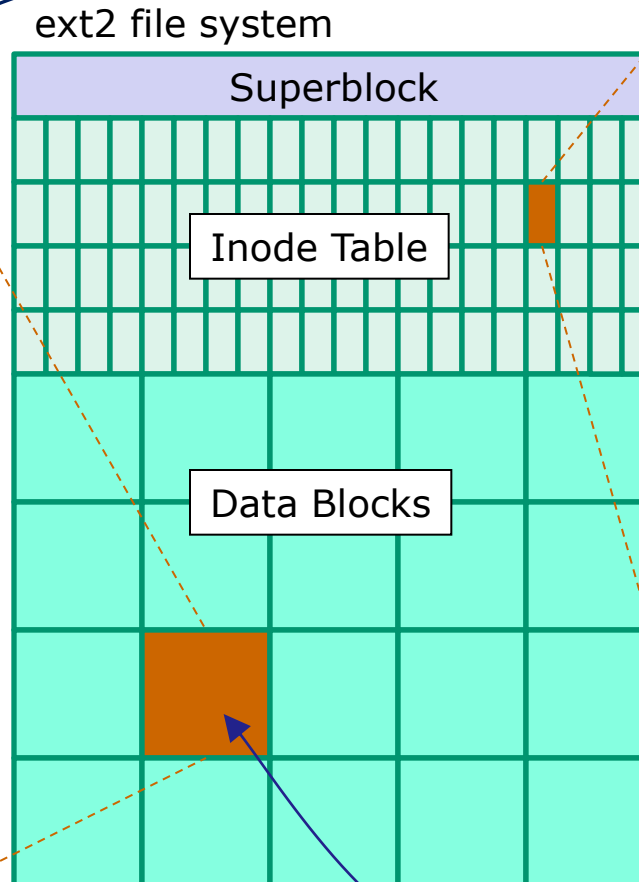
Now I don't want this to scare you, but my bunk mate has malaria. You remember Jeffrey Hardy? Their about to organize a searching party.

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

```
[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 ¹¹⁰¹⁰⁰¹⁰⁰rw-r--r-- or 644

owner has read and write

group has only read

others have only read

numeric form

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 `rwXrw-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?

1	1	0	1	0	0	1	0	0
r	w	-	r	-	-	r	-	-

Answer:

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

File Permissions

Does the simben90 user have read access to /etc/samba/smb.conf?

File Permissions

Does the simben90 user have read access to /etc/samba/smb.conf?

Answer: yes

```
/home/cis90/simben $ 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 – change owner of a file

```
[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 the **chown** command*

chgrp

File Permissions

chgrp – change group a file belongs to

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

*change group to
users*

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

*change group back
to cis90*

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

chmod
(mnemonic)

File Permissions

mnemonics – “memory aids”

Examples:

u+w = add write permission to user

u-w = remove write permission from user

u+wx = add write and execute permission to user

g+r = add read permission to group

g-rwx – remove read, write, execute permissions
from group

o+rw = add read, write permissions to others

o-r = remove read permission from others

+x = add execute permission to user, group and
others

+rw = add read & write permissions to user, group
and others

uo+w = add write permission to user and others

u+rwx,o-rwx = add read, write, execute
permissions to user but remove them from others

Definitions:

u=user (owner)

g=group

o=other

r=read permission

w=write permission

x=execute permission

*combinations allowed but
no blanks around
commas please*

File Permissions

chmod command – change permissions (mnemonic method)

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

*add execute
permission for
owner*

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

*add execute
permission for
group*

Use chmod to add or remove permissions from a file

File Permissions

chmod command – change permissions (mnemonic method)

```
[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

chmod (numerical)

File Permissions

chmod command – change permissions (numerical method)

```
[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 – change permissions (numerical method)

```
[simmsben@opus Directory3]$ chmod 777 myfile  
[simmsben@opus Directory3]$ ls -l myfile  
-rw-rwxrwx 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

Read

Write

Execute

File Permissions

Commands that require file permissions



```
inodeNum1 fileName1
inodeNum2 fileName2
:
:
```

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

read permission is required whenever file contents must be accessed

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



```
inodeNum1 fileName1
inodeNum2 fileName2
:
:
```

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

write permission is required whenever file contents are written

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



```
inodeNum1 fileName1
inodeNum2 fileName2
:
:
```

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

execute permission is required to load and run a file

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

Directory Write Permission



```
inodeNum1 fileName1
inodeNum2 fileName2
:                :
```

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

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

Directory with no write permission example 1

```
[simben@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
```

Benji 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
```

Benji (and everyone else) 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 containing the file.

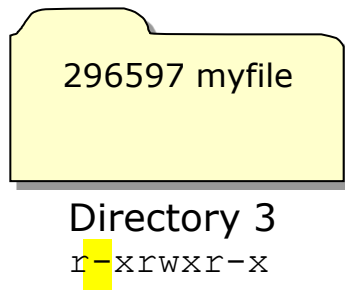
This is so you can write the revised file contents (with the file removed) to the directory. Remember that directories are like phone books and only contain file names and inode numbers.

The permissions on the file being removed 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



Owner tries to write revised file contents to Directory3

Permission denied

Directory with write permission 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 Benji has
no permissions
on this file*

```
[simmsben@opus Directory3]$ rm myfile
```

```
rm: remove write-protected regular empty file `myfile'? yes
```

```
[simmsben@opus Directory3]$
```

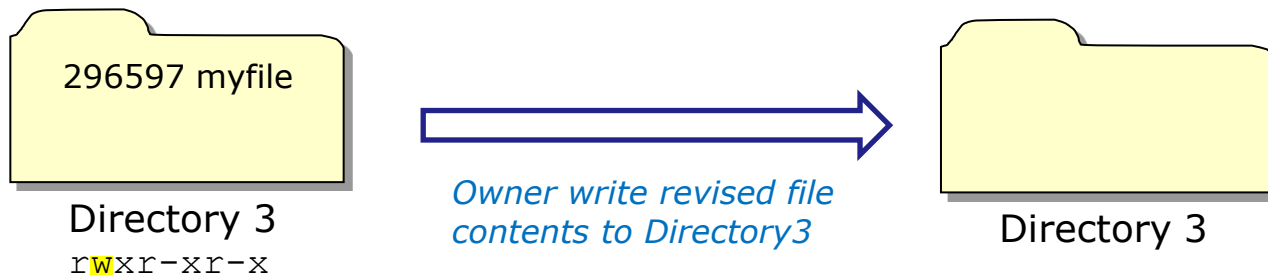
So how come he can delete it?



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.

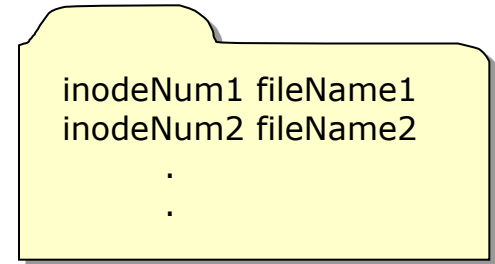


Directory permissions READ

Directory Read Permission



rwx



rwx

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

*Use the **ls** command to read the contents of a directory.
Note, having read permission is required!*

Directory Read Permission

r w x read write execute	r w x read write execute	r w x read write execute
user (owner)	group	others

Start with normal directory permissions:

```
/home/cis90/roddyduk $ ls -ld examples/
drwxrwxr-x 5 roddyduk cis90 4096 Oct 19 13:49 examples/
```

```
/home/cis90/roddyduk $ ls -i examples/
2525532 birds 2525533 dogs
```

2525532 birds
2525533 dogs

examples

If read permission is removed from the directory ...

Can we still list the directory contents?

Directory Read Permission

r w x read write execute	r w x read write execute	r w x read write execute
user (owner)	group	others

Remove read permission and confirm it's gone

```
/home/cis90/roddyduk $ chmod u-r examples
/home/cis90/roddyduk $ ls -ld examples
d-wxrwxr-x 4 roddyduk cis90 4096 Oct 19 13:59 examples
```

2525532 birds
2525533 dogs

examples

Can we still list the directory contents?

```
/home/cis90/roddyduk $ ls -l examples/
ls: examples/: Permission denied
/home/cis90/roddyduk $
```

NO!

Directory Read Permission

r w x read write execute	r w x read write execute	r w x read write execute
user (owner)	group	others

Start with normal directory permissions:

```
/home/cis90/roddyduk $ ls -ld examples/
drwxrwxr-x 5 roddyduk cis90 4096 Oct 19 13:49 examples/
```

```
/home/cis90/roddyduk $ ls -li examples/
2525532 birds 2525533 dogs
```

2525532 birds
2525533 dogs

examples

If read permission is removed from the directory ...

*Can we still **cd** into the directory?*

Directory Read Permission

r	w	x	r	w	x	r	w	x
read	write	execute	read	write	execute	read	write	execute
user			group			others		

Remove read permission and confirm it's gone

```
/home/cis90/roddyduk $ chmod u-r examples
/home/cis90/roddyduk $ ls -ld examples
d-wxrwxr-x 4 roddyduk cis90 4096 Oct 19 13:59 examples
```

2525532 birds
2525533 dogs

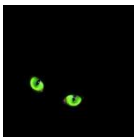
examples

*Can we still **cd** into the directory?*

```
/home/cis90/roddyduk $ cd examples/
/home/cis90/roddyduk/examples $ ls
ls: .: Permission denied
/home/cis90/roddyduk/examples $ ls birds
abby nibbie
```

Yes, but ...

- *we still can't list the contents,*
- *yet we can still access anything in the directory!*



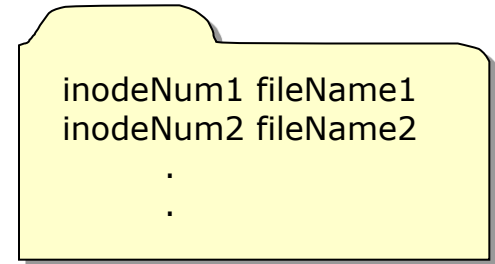
It's like walking into a pitch black room. You can't see anything, but if you know where things are you can still use them.

Directory permissions WRITE (more)

Directory Write Permission



rwX



rwX

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

Removing directory w permission

- can't cp files to it,
- can't remove files,
- can't move files out,
- can't add links

Directory Read Permission

r w x read write execute	r w x read write execute	r w x read write execute
user (owner)	group	others

Start with normal directory permissions:

```
/home/cis90/roddyduk $ ls -ld examples/
drwxrwxr-x 5 roddyduk cis90 4096 Oct 19 13:49 examples/
```

```
/home/cis90/roddyduk $ ls -li examples/
2525532 birds 2525533 dogs
```

2525532 birds
2525533 dogs

examples

If write permission is removed from the directory ...

Can we remove files from the directory?

Directory Read Permission

r w x read write execute	r w x read write execute	r w x read write execute
user (owner)	group	others

Remove write permission and confirm it's gone

```
/home/cis90/roddyduk $ chmod u-w examples
/home/cis90/roddyduk $ ls -ld examples
dr-xrwxr-x 4 roddyduk cis90 4096 Oct 19 13:59 examples/
```

2525532 birds
2525533 dogs

examples

Can we remove files from the directory?

```
/home/cis90/roddyduk/examples $ rmdir dogs
rmdir: dogs: Permission denied
```

NO!

Yet we can cd into and list directory contents

```
/home/cis90/roddyduk $ cd examples/
/home/cis90/roddyduk/examples $ ls
birds  dogs
```

Directory Read Permission

r w x read write execute	r w x read write execute	r w x read write execute
user (owner)	group	others

Start with normal directory permissions:

```
/home/cis90/roddyduk $ ls -ld examples/
drwxrwxr-x 5 roddyduk cis90 4096 Oct 19 13:49 examples/
```

```
/home/cis90/roddyduk $ ls -li examples/
2525532 birds 2525533 dogs
```

2525532 birds
2525533 dogs

examples

If write permission is removed from the directory ...

Can we create new files or copy/move files into the directory?

Directory Read Permission

r w x read write execute	r w x read write execute	r w x read write execute
user (owner)	group	others

Remove write permission and confirm it's gone

```
/home/cis90/roddyduk $ chmod u-w examples
/home/cis90/roddyduk $ ls -ld examples
dr-xrwxr-x 4 roddyduk cis90 4096 Oct 19 13:59 examples/
```

2525532 birds
2525533 dogs

examples

Can we create new files or copy/move files into the directory?

```
/home/cis90/roddyduk $ cp letter examples/
cp: cannot create regular file `examples/letter': Permission denied
/home/cis90/roddyduk $ mv letter examples/
mv: cannot move `letter' to `examples/letter': Permission denied
/home/cis90/roddyduk $ touch examples/newfile
touch: cannot touch `examples/newfile': Permission denied
/home/cis90/roddyduk $
```

NO!

To change the contents of a directory (either add or remove files) requires write permission

Directory Read Permission

r w x read write execute	r w x read write execute	r w x read write execute
user (owner)	group	others

Start with normal directory permissions:

```
/home/cis90/roddyduk $ ls -ld examples/
drwxrwxr-x 5 roddyduk cis90 4096 Oct 19 13:49 examples/
```

```
/home/cis90/roddyduk $ ls -li examples/
2525532 birds 2525533 dogs
```

2525532 birds
2525533 dogs

examples

If write permission is removed from the directory ...

Can we move files out of the directory?

Directory Read Permission

r w x read write execute	r w x read write execute	r w x read write execute
user (owner)	group	others

Remove write permission and confirm it's gone

```
/home/cis90/roddyduk $ chmod u-w examples
/home/cis90/roddyduk $ ls -ld examples
dr-xrwxr-x 4 roddyduk cis90 4096 Oct 19 13:59 examples/
```

2525532 birds
2525533 dogs

examples

Can we move files out of the directory?

```
/home/cis90/roddyduk $ mv examples/birds .
mv: cannot move `examples/birds' to `./birds': Permission denied
```

NO!

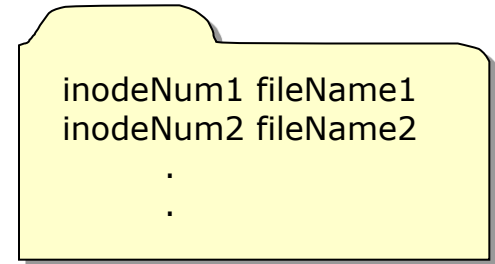
To change the contents of a directory (either add or remove files) requires write permission

Directory permissions EXECUTE

Directory Execute Permission



rwX



rwX

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

Removing directory x permission

- **cannot** retrieve inode information (*ls -l*)
(which means no file data either)
- **cannot** cd into directory

Directory Read Permission

r w x read write execute	r w x read write execute	r w x read write execute
user (owner)	group	others

Start with normal directory permissions:

```
/home/cis90/roddyduk $ ls -ld examples/  
drwxrwxr-x 5 roddyduk cis90 4096 Oct 19 13:49 examples/
```

```
/home/cis90/roddyduk $ ls -li examples/  
2525532 birds 2525533 dogs
```

2525532 birds
2525533 dogs

examples

If execute permission is removed from the directory ...

Can we change into (cd) the directory?

Directory Read Permission

r w x read write execute	r w x read write execute	r w x read write execute
user (owner)	group	others

Remove execute permission and confirm it's gone

```
/home/cis90/roddyduk $ chmod u-x examples
/home/cis90/roddyduk $ ls -ld examples
drw-rwxr-x 4 roddyduk cis90 4096 Oct 19 13:59 examples/
```

2525532 birds
2525533 dogs

examples

Can we change into (cd) the directory?

```
/home/cis90/roddyduk $ cd examples/
-bash: cd: examples/: Permission denied
/home/cis90/roddyduk $
```

NO!

Execute permission is required to change into a directory or to get inode based information for any of the files in the directory. Note, without inode information you can't get to a file's data.

Directory Read Permission

r w x read write execute	r w x read write execute	r w x read write execute
user (owner)	group	others

Start with normal directory permissions:

```
/home/cis90/roddyduk $ ls -ld examples/
drwxrwxr-x 5 roddyduk cis90 4096 Oct 19 13:49 examples/
```

```
/home/cis90/roddyduk $ ls -li examples/
2525532 birds 2525533 dogs
```

2525532 birds
2525533 dogs

examples

If execute permission is removed from the directory ...

Can we list directory contents?

Directory Read Permission

r w x read write execute	r w x read write execute	r w x read write execute
user (owner)	group	others

Remove execute permission and confirm it's gone

```
/home/cis90/roddyduk $ chmod u-x examples
/home/cis90/roddyduk $ ls -ld examples
drw-rwxr-x 4 roddyduk cis90 4096 Oct 19 13:59 examples/
```

2525532 birds
2525533 dogs

examples

Can list directory contents?

```
/home/cis90/roddyduk $ ls examples/
birds  dogs
```

Yes

Directory Read Permission

r w x read write execute	r w x read write execute	r w x read write execute
user (owner)	group	others

Start with normal directory permissions:

```
/home/cis90/roddyduk $ ls -ld examples/
drwxrwxr-x 5 roddyduk cis90 4096 Oct 19 13:49 examples/
```

```
/home/cis90/roddyduk $ ls -i examples/
2525532 birds 2525533 dogs
```

2525532 birds
2525533 dogs

examples

If execute permission is removed from the directory ...

Can we do a long listing of the directory?

Directory Read Permission

r w x read write execute	r w x read write execute	r w x read write execute
user (owner)	group	others

Remove execute permission and confirm it's gone

```
/home/cis90/roddyduk $ chmod u-x examples
/home/cis90/roddyduk $ ls -ld examples
drw-rwxr-x 4 roddyduk cis90 4096 Oct 19 13:59 examples/
```

2525532 birds
2525533 dogs

examples

Can we do a long listing (show inode information) of the directory?

```
/home/cis90/roddyduk $ ls -l examples/
total 0
?----- ? ? ? ?
?----- ? ? ? ?
```

? **birds**
? **dogs**

Incomplete! Only file names. No information kept in the file's inode is shown!

We can read the filenames, but without execute permission we can't retrieve information from the inode

umask

File Permissions

Default Permissions

What permissions are files created with?

Answer: This is determined by the umask setting

Default system permissions

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

The user's 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 With no argument, 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 system permissions for a file</i>
002	-----w-	<i>umask setting (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 system permissions for a file</i>
000	-----	<i>umask setting (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- *default system permissions for a file*

022 ----w--w- *umask setting (strips these permissions from default)*

644 **rw-r--r--** *result after masking*

When new files are created

```
/home/cis90/roddyduk $ touch mydogs  
/home/cis90/roddyduk $ ls -l mydogs  
-rw-rw-r-- 1 roddyduk cis90 0 Oct 19 13:16 mydogs
```

When a new file is created:

- the permissions are based on the umask value
- the owner is set to the user creating the file
- the group is set to the user's primary group

permissions fun

Go slowly and follow
all directions

Permissions Exercise

Find the hidden treasure trove



- Find the buried treasure in your Hidden folder.
- Beware! - once you find it, make sure you set permissions to protect your treasure from *everyone!*

Lab 6



Lab 6: File Permissions

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

Browse to: <http://opus.cabrillo.edu/forum/viewforum.php?f=46>

Check the forum for any late breaking news about this lab. The forum is also the place to go if you get stuck, have a question or want to share something you have learned about this lab.

Procedure

Log on to Opus 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.

Part I - Making Directories

1. From your home directory, do a long listing with 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:
chmod -r uhistory
Try to display the contents of the file as you did above. Does it work?
5. Now give read permission back but take away write permission:
chmod 444 uhistory
Verify the success of the above command.
6. Take away execute (search) permission from the **misc** directory:
chmod -x misc
Do short and long listings of the **misc** directory using the **ls** and **ls -l** commands.

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

***Be sure and finish
Lab 5 before
starting Lab 6!***

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 `rwxr-xr--` ?
- Does **chmod o+w** give write permission to the *owner* or to *other* users?

Backup

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	TANF/CALWORKS
☐ No	
☐ Yes	SSI (Supplemental Security Income)
☐ No	
☐ Yes	GA (General Assistance)
☐ No	
☐ Yes	Does your <u>income</u> qualify you for a fee waiver?
☐ No	
☐ Yes	Are you a single parent with custody of one or more minor children?
☐ No	
☐ Yes	Are you a <u>displaced homemaker</u> attending Cabrillo to develop job skills?
☐ No	
☐ Yes	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?
☐ No	

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

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 ~]$
```

