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 -

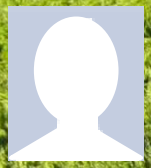
- CCC Confer wall paper & quiz -

- Set up Polycom phone/extension mics -
- Wireless lapel mic backup battery -
- Backup slides, CCC info, handouts on flash drive -

CIS 90 - Lesson 7

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


Sean C.

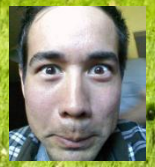

Donald

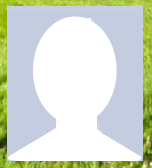

Carlile


Andrew


Sean Fa.


Carter

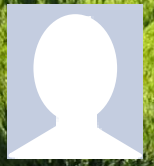

Sean Fy.


Dajan

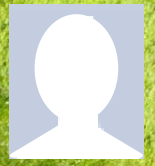

Bryn

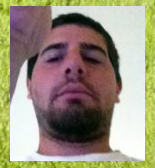

Rita

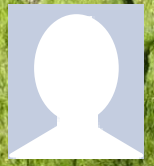

Kelly

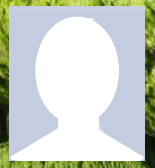

Ben


Ray


Fidel

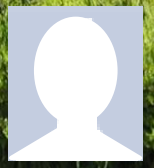

Michael

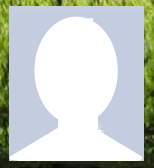

Evan


Josh


Carlos


Gustavo


Jessica


Evie


Jacob


Humberto


Chad

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

Quiz

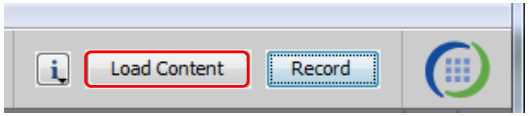
Please answer these questions **in the order** shown:

See electronic white board

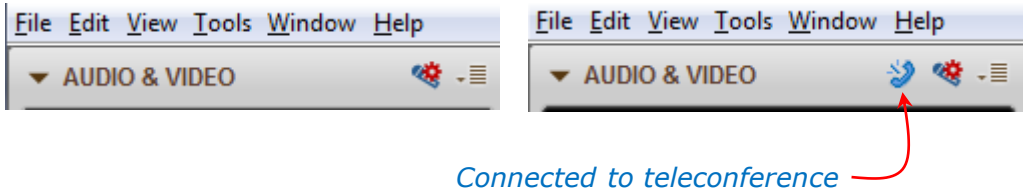
email answers to: risimms@cabrillo.edu

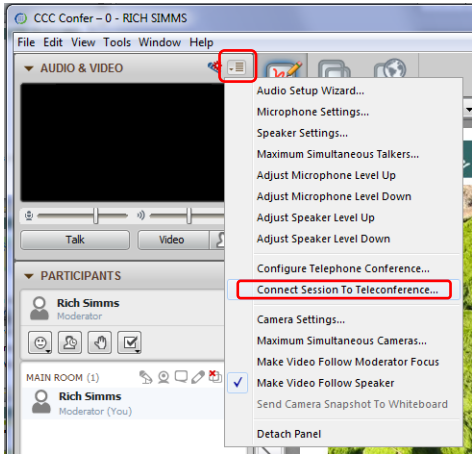
(answers must be emailed within the first few minutes of class for credit)

[ ] Load White Board with *cis*lesson??*-WB*

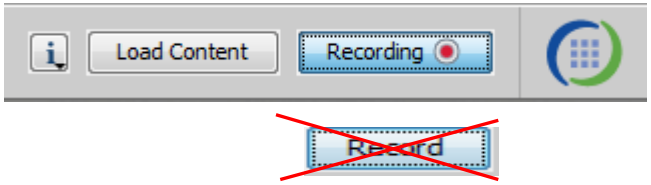


[ ] Connect session to Teleconference





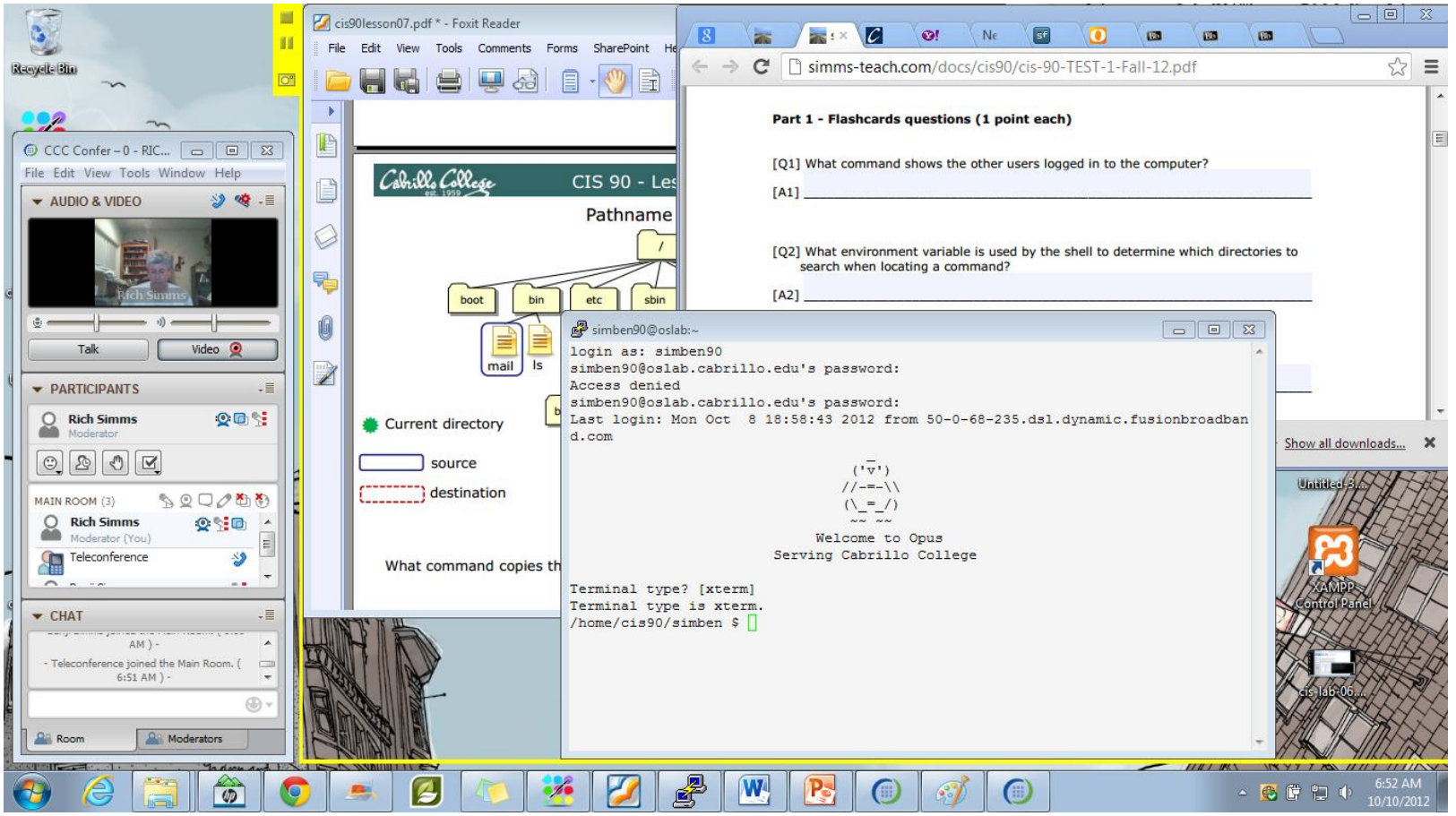
[ ] Is recording on?



[ ] Toggle Talk button to not use Mic

CCC Confer

[ ] Video (webcam) optional



File Permissions

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

Questions

Previous material and assignment

1. Questions on Test #1?
   - graded tests in your home directory
   - answers in /home/cis90/answers
2. Questions on last lesson?
3. Questions on Lab 5?

Test 1 Post Mortem

Test 1 – Results

Missed Q12 = 13	*directory pathnames*
Missed Q8 = 12	*pathnames and inodes*
Missed Q32 = 12	*special prompts*
Missed Q24 = 11	*parsing*
Missed Q33 = 10	*/etc/passwd fields*
Missed Q31 = 9	*how shell works*
Missed Q20 = 8	*relative pathname*
Missed Q27 = 7	*basic file types*
Missed Q21 = 7	*sorted listings*
Missed Q19 = 7	*head command*
Missed Q18 = 7	*/etc/passwd fields*
Missed Q10 = 7	*sorting files by size*
Missed Q29 = 6	
Missed Q26 = 6	
Missed Q22 = 6	
Missed Q11 = 6	
Missed Q7 = 5	
Missed Q25 = 5	
Missed Q23 = 5	
Missed Q30 = 4	
Missed Q28 = 4	
Missed Q2 = 4	
Missed Q17 = 4	
Missed Q9 = 3	*/etc/passwd giveaway*
Missed Q5 = 3	*relative pathname*
Missed Q13 = 3	*man pages*
Missed Q6 = 2	*prompt string*
Missed Q15 = 2	*bc and mails*
Missed Q16 = 1	*command inputs*
Missed Q4 = 0	*xxd command*
Missed Q3 = 0	*chat command*
Missed Q14 = 0	*reading mail*
Missed Q1 = 0	*who command*

Correct answers are in /home/cis90/answers

From Lesson 4

Question: What is the absolute pathname of /etc/passwd?

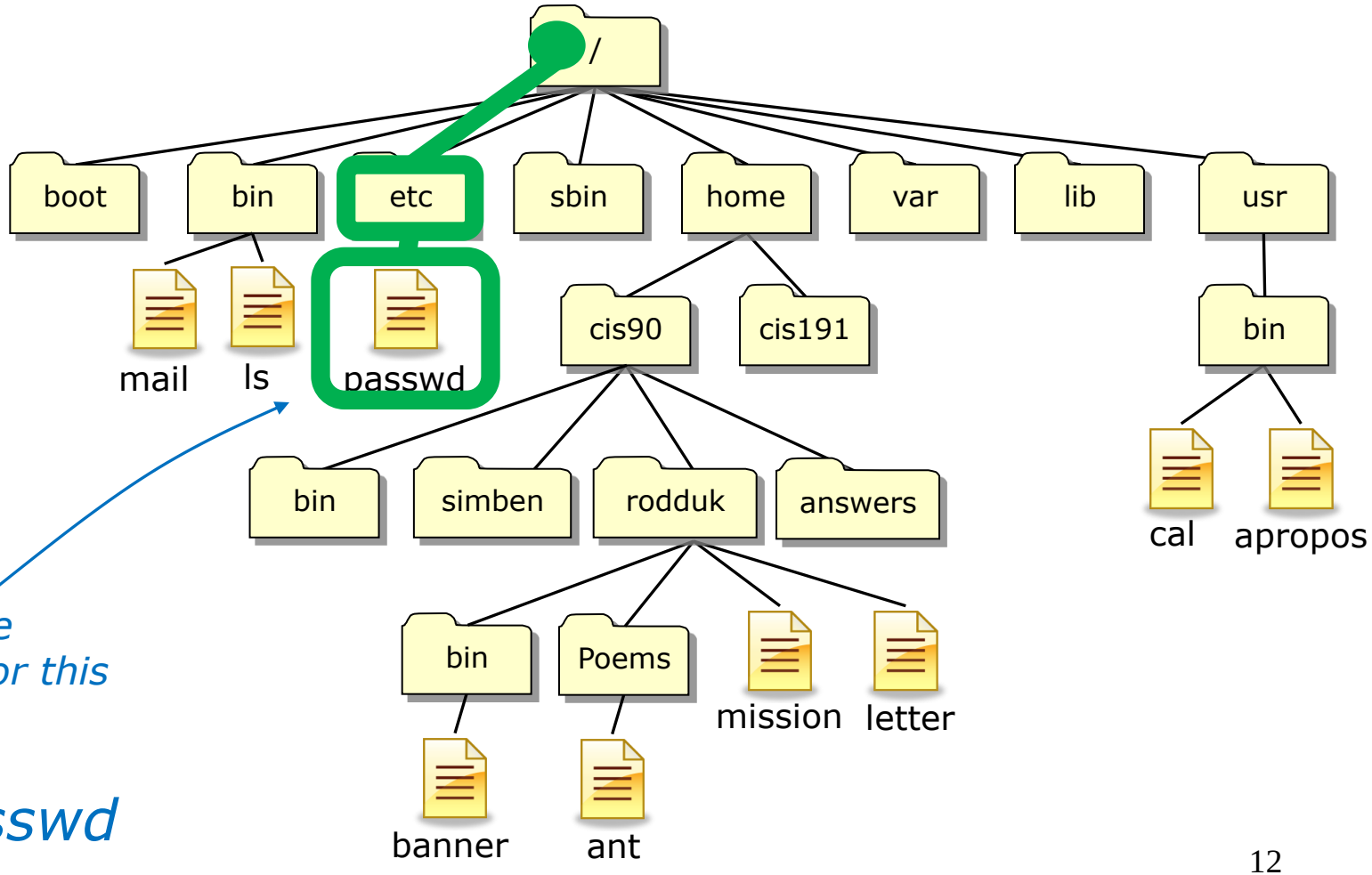
Answer: /etc/passwd

This is the "give away" question that I put on each test till we get 100% correct responses

… not yet with Test 1 results!

UNIX File Tree

/ = root of the tree

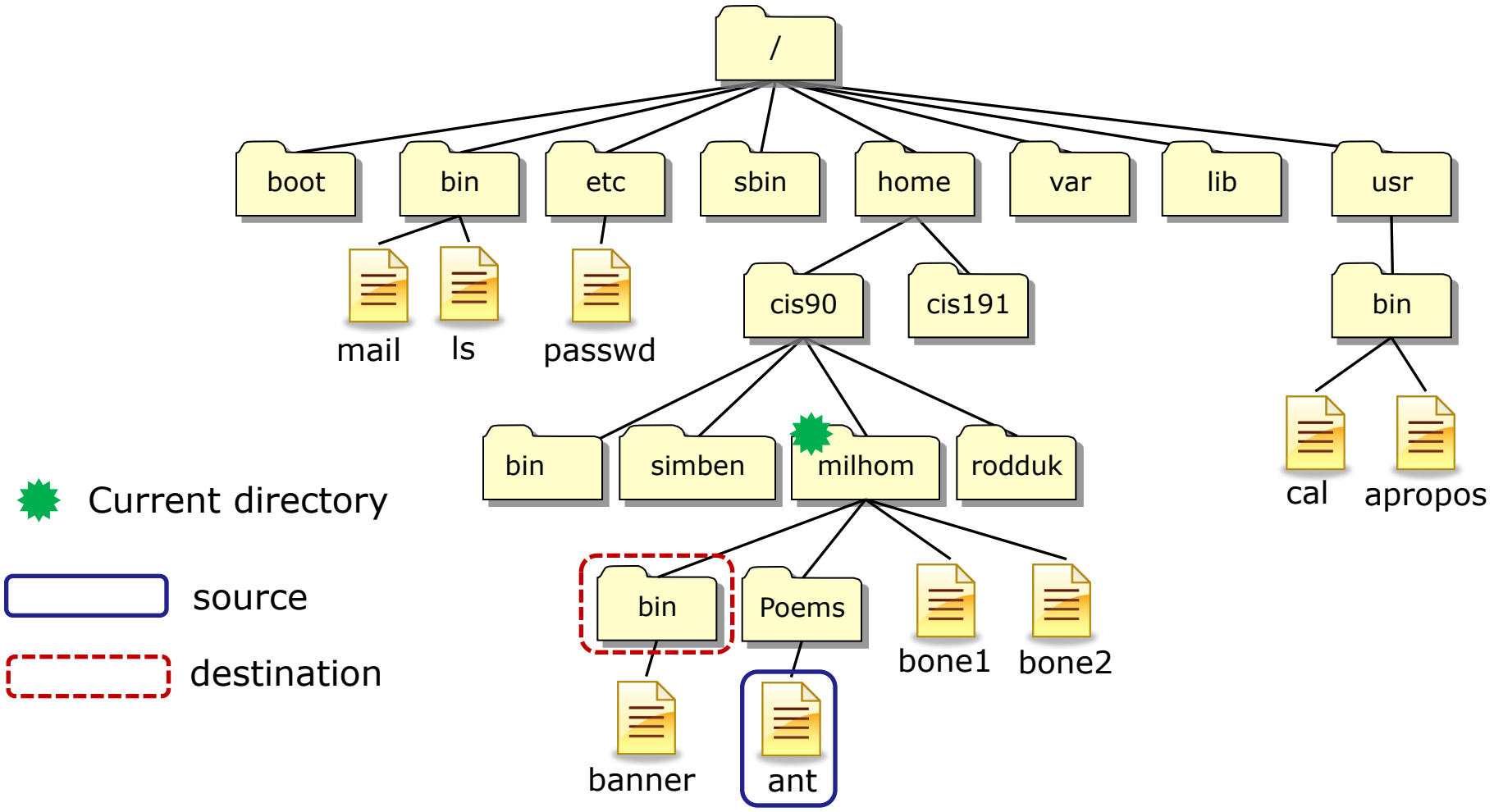


The absolute pathname for this file is:

/etc/passwd

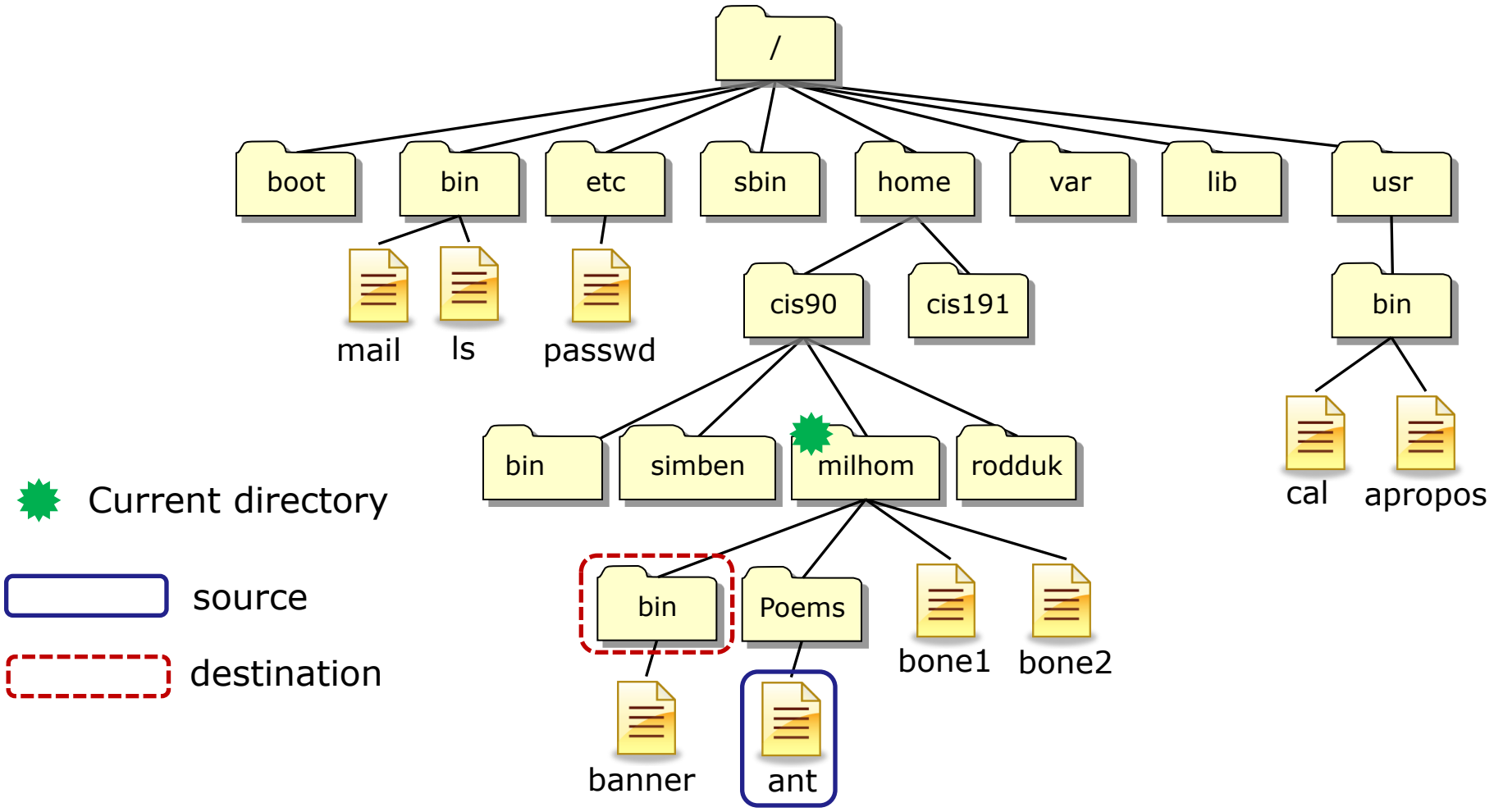
Review

Pathname Practice



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

Pathname Practice



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

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

Parse: **`cp Poems/ant bin/`**

Parse: `cp Poems/ant bin/`

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

Shell parses this command line

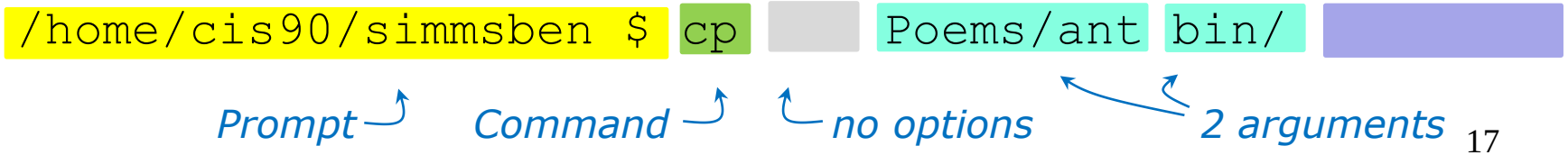
*The **command** will be loaded only if the shell can locate it on your path (defined by the PATH variable)*

***Options** modify the behavior of the command*

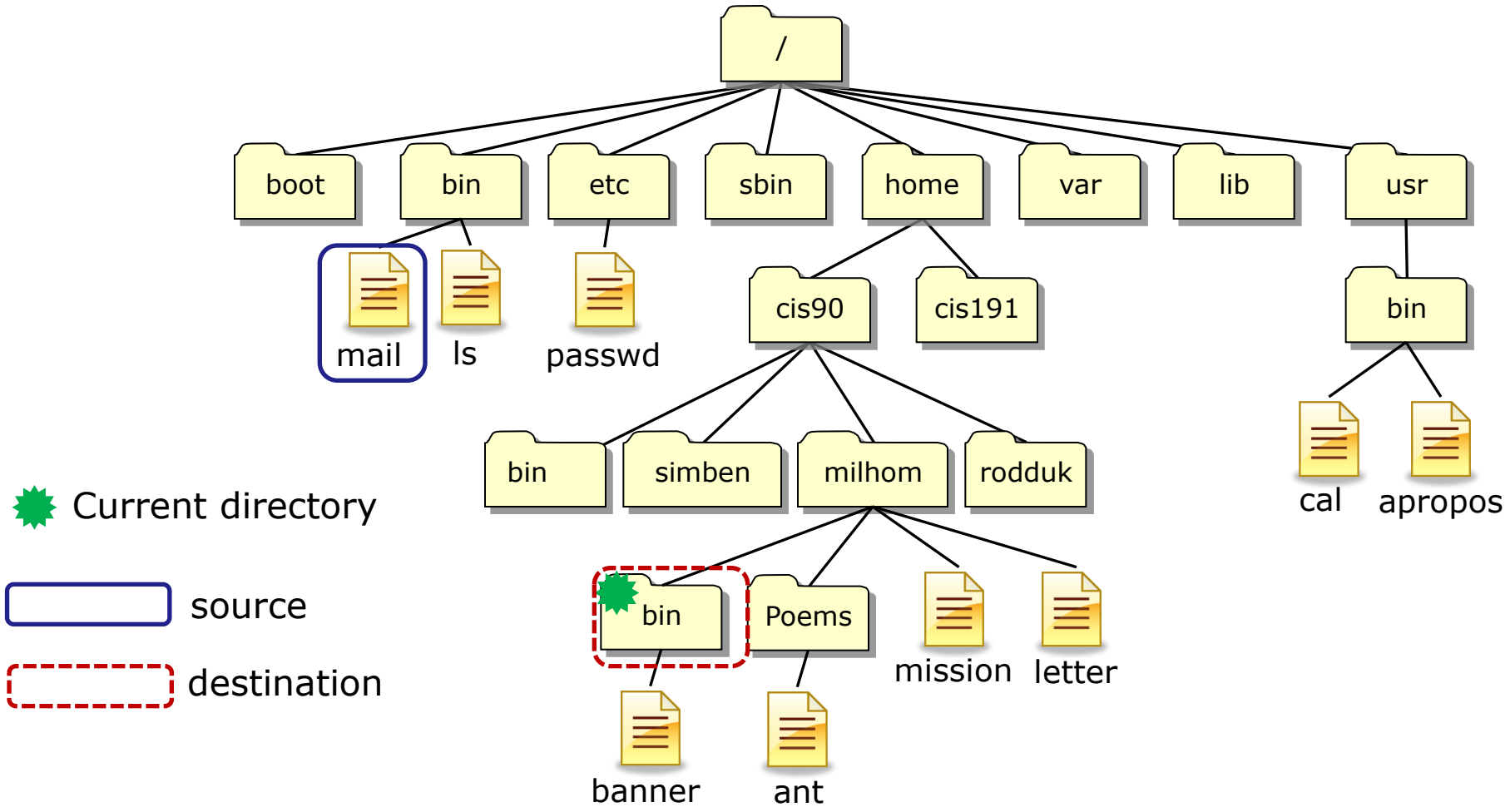
***Arguments** are what the command works upon*

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

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

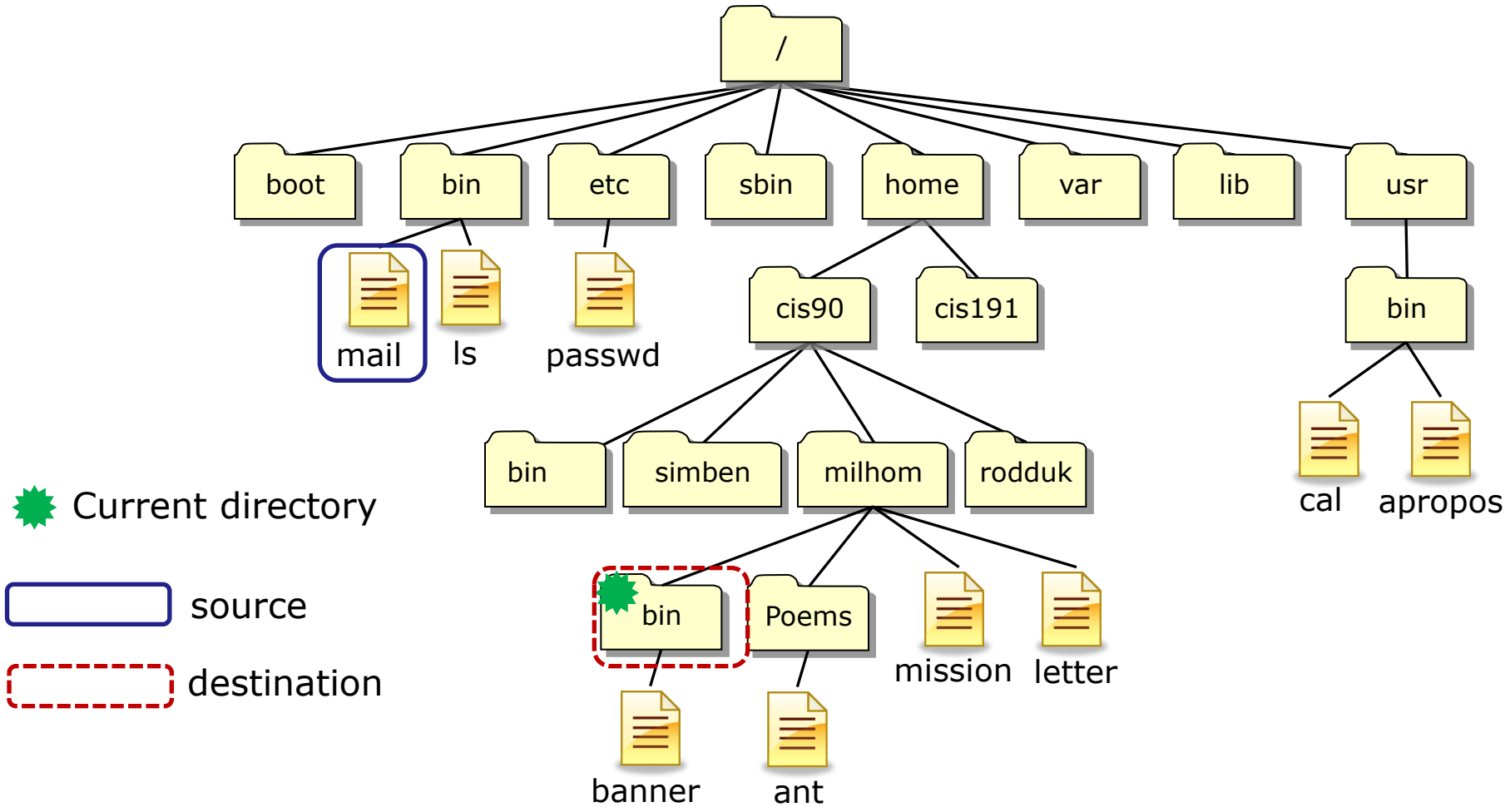


Pathname Practice



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

Pathname Practice



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

```
cp /bin/mail .
```

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

Parse: `cp /bin/mail .`

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

Shell parses this command line

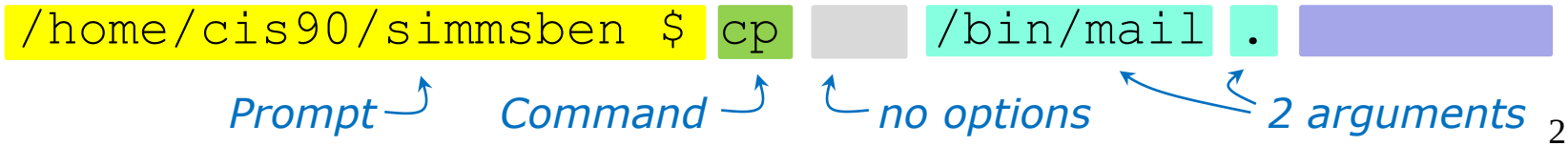
*The **command** will be loaded only if the shell can locate it on your path (defined by the PATH variable)*

***Options** modify the behavior of the command*

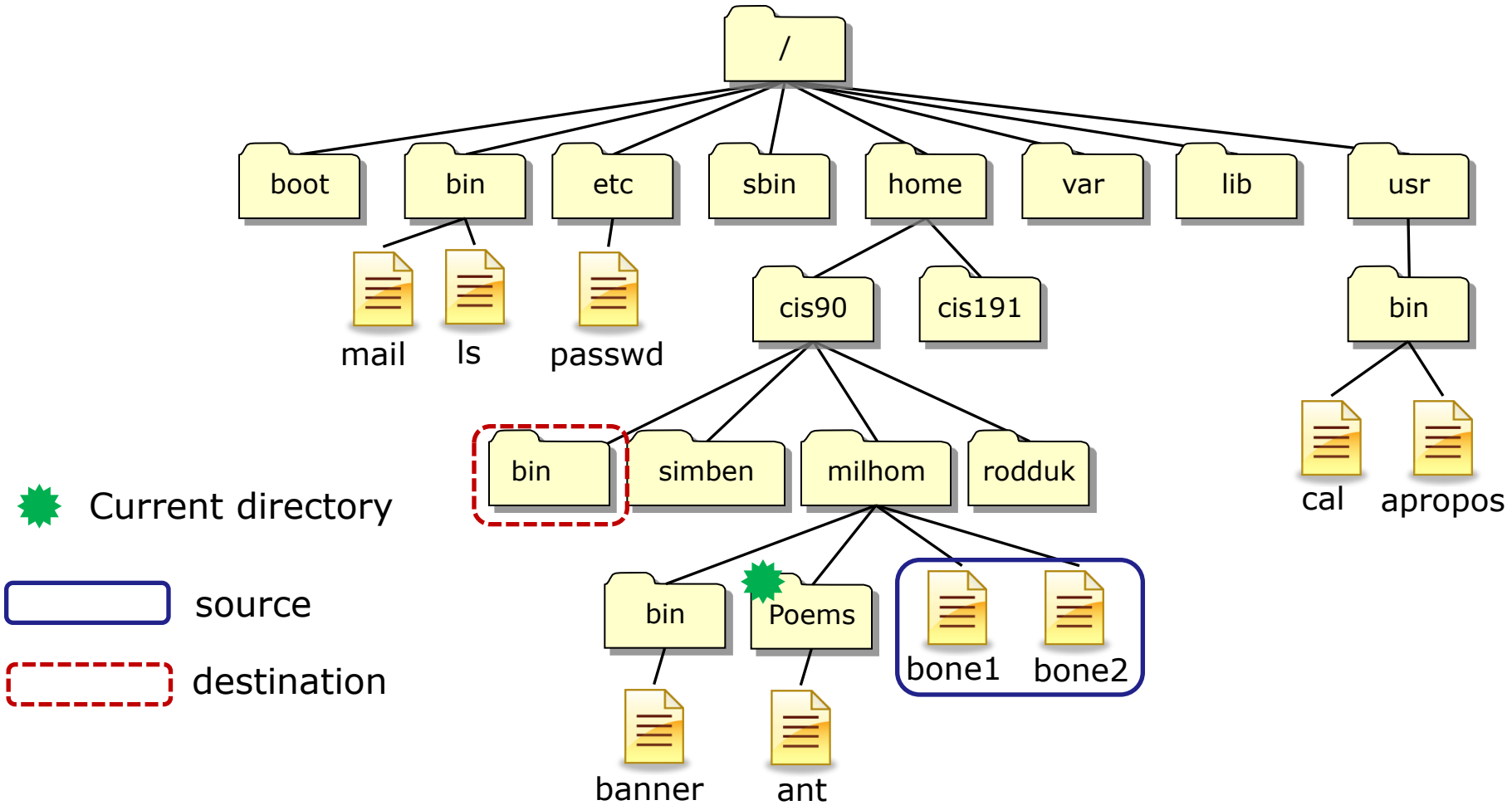
***Arguments** are what the command works upon*

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

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

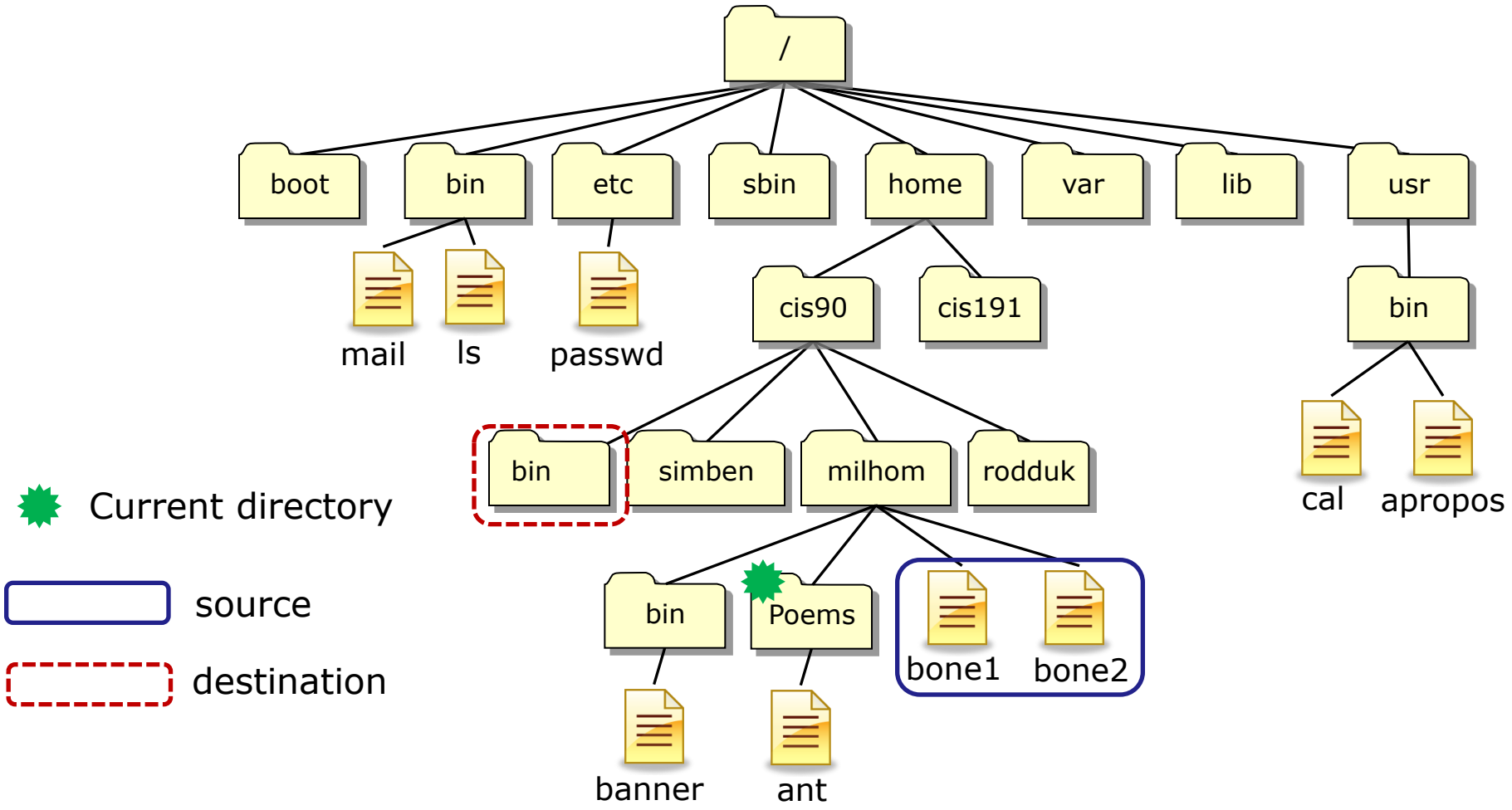


Pathname Practice



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

Pathname Practice



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

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

Parse: **`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*

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

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

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

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

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

Prompt *Command* *no options* *3 arguments*

Housekeeping

1) Lab 5 is due tonight at 11:59PM

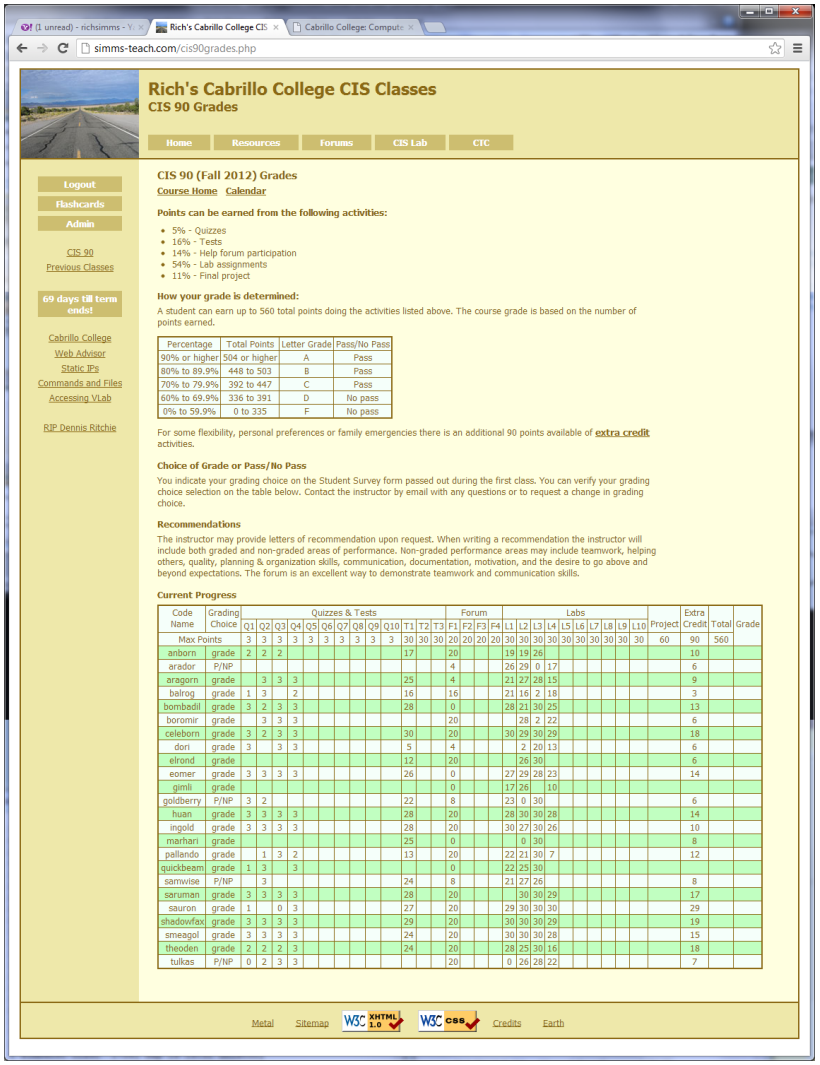
2) A **check5** script is available (see forum)

3) Next five forum posts due next week

Tip:

- Review graded work in your home directory
- Move graded work to your *class/labs* or *class/exams* directories
- Compare your answers to quizzes, tests and labs with those in */home/cis90/answers*

Housekeeping



Please monitor your grades on the Grades web page.

You can also use Jesse's ***checkgrades*** *script on Opus and provide your code name as an argument.*

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

Bi-annual Campus Climate Student Survey

https://www.surveymonkey.com/s/StudentCampusClimateSurvey2012

This survey will take approximately 15 minutes for students to complete online. If you'd like students to get credit – or extra credit - for completing the survey, Judy will provide names/sections of respondents to you at the end of October. It is otherwise considered optional and voluntary, as there is no "captive audience" online, as we have in classrooms, but it is exceedingly important that we get a good response rate of the student body, overall.

Three points extra credit if I get your name (not your survey answers) from Judy at the end of the month.

Permissions

R=Read
W=Write
X=Execute

File Permissions

File permissions are used to control access to files and directories

There are three basic permissions: **read, write and execute**

Which can be applied to:

1) The **user** (**owner**) of the file
2) A **group** of users
3) Everyone else (**others**)

File Permissions

Interpreting the permission codes on long listings

simben90@oslab:~

```
/home/cis90/simben $ ls -l
total 472
-rw-rw-r--. 1 simben90 cis90   4008 Sep 11 22:23 archives
-rw-r--r--. 6 rsimms   cis90  10576 Aug  1 18:49 bigfile
drwxr-xr-x. 2 simben90 cis90   4096 Oct  5 10:25 bin
drwxrwxr-x. 4 simben90 cis90   4096 Oct  5 10:21 class
-rw-------. 1 simben90 cis90   1894 Sep 20 06:23 dead.letter
drwxrwxr-x. 2 simben90 cis90   4096 Oct  5 10:25 docs
drwxrwxr-x. 2 simben90 cis90   4096 Oct  5 10:30 edits
drwxrwxr-x. 2 simben90 cis90   4096 Oct  5 10:41 etc
d---------. 2 simben90 cis90   4096 Feb  1  2002 Hidden
-r--------. 1 simben90 staff   2780 Sep  6 13:47 lab01.graded
-r--------. 1 simben90 staff   1312 Sep 13 12:27 lab02.graded
-r--------. 1 simben90 staff    814 Sep 27 13:08 lab04.graded
-rw-r--r--. 1 simben90 cis90   1059 Oct  7 14:41 letter
-rw-r--r--. 1 simben90 cis90    208 Oct  5 10:45 log
-rwxr-xr-x. 1 simben90 cis90 375252 Oct  7 14:05 mail
-rw-rw-r--. 1 simben90 cis90   3766 Sep 12 18:53 mbox
drwxr-xr-x. 2 simben90 cis90   4096 Oct  5 10:30 misc
drwxr-xr-x. 7 simben90 cis90   4096 Oct  5 10:35 poems
-r--------. 1 simben90 staff   5899 Oct  4 11:04 test01.graded
-rw-rw-r--. 1 simben90 cis90  17341 Sep 19 19:31 uhistory
/home/cis90/simben $
```

Use long listings to view file permissions

File Permissions

Interpreting the permission codes on long listings

simben90@oslab:~

```
/home/cis90/simben $ ls -l
total 472
-rw-rw-r--. 1 simben90 cis90   4008 Sep 11 22:23 archives
-rw-r--r--. 6 rsimms   cis90  10576 Aug  1 18:49 bigfile
drwxr-xr-x. 2 simben90 cis90   4096 Oct  5 10:25 bin
drwxrwxr-x. 4 simben90 cis90   4096 Oct  5 10:21 class
-rw-------. 1 simben90 cis90   1894 Sep 20 06:23 dead.letter
drwxrwxr-x. 2 simben90 cis90   4096 Oct  5 10:25 docs
drwxrwxr-x. 2 simben90 cis90   4096 Oct  5 10:30 edits
drwxrwxr-x. 2 simben90 cis90   4096 Oct  5 10:41 etc
d---------. 2 simben90 cis90   4096 Feb  1  2002 Hidden
-r--------. 1 simben90 staff   2780 Sep  6 13:47 lab01.graded
-r--------. 1 simben90 staff   1312 Sep 13 12:27 lab02.graded
-r--------. 1 simben90 staff    814 Sep 27 13:08 lab04.graded
-rw-r--r--. 1 simben90 cis90   1059 Oct  7 14:41 letter
-rw-r--r--. 1 simben90 cis90    208 Oct  5 10:45 log
-rwxr-xr-x. 1 simben90 cis90 375252 Oct  7 14:05 mail
-rw-rw-r--. 1 simben90 cis90   3766 Sep 12 18:53 mbox
drwxr-xr-x. 2 simben90 cis90   4096 Oct  5 10:30 misc
drwxr-xr-x. 7 simben90 cis90   4096 Oct  5 10:35 poems
-r--------. 1 simben90 staff   5899 Oct  4 11:04 test01.graded
-rw-rw-r--. 1 simben90 cis90  17341 Sep 19 19:31 uhistory
/home/cis90/simben $
```

Columns 2-10 show the permissions on each file using:
***r** (read), **w** (write), **x** (execute) or **–** (no permission)*

File Permissions

Interpreting the permission codes on long listings

simben90@oslab:~

```
/home/cis90/simben $ ls -l
total 472
-rw-rw-r--. 1 simben90 cis90   4008 Sep 11 22:23 archives
-rw-r--r--. 6 rsimms   cis90  10576 Aug  1 18:49 bigfile
drwxr-xr-x. 2 simben90 cis90   4096 Oct  5 10:25 bin
drwxrwxr-x. 4 simben90 cis90   4096 Oct  5 10:21 class
-rw-------. 1 simben90 cis90   1894 Sep 20 06:23 dead.letter
drwxrwxr-x. 2 simben90 cis90   4096 Oct  5 10:25 docs
drwxrwxr-x. 2 simben90 cis90   4096 Oct  5 10:30 edits
drwxrwxr-x. 2 simben90 cis90   4096 Oct  5 10:41 etc
d---------. 2 simben90 cis90   4096 Feb  1  2002 Hidden
-r--------. 1 simben90 staff   2780 Sep  6 13:47 lab01.graded
-r--------. 1 simben90 staff   1312 Sep 13 12:27 lab02.graded
-r--------. 1 simben90 staff    814 Sep 27 13:08 lab04.graded
-rw-r--r--. 1 simben90 cis90   1059 Oct  7 14:41 letter
-rw-r--r--. 1 simben90 cis90    208 Oct  5 10:45 log
-rwxr-xr-x. 1 simben90 cis90 375252 Oct  7 14:05 mail
-rw-rw-r--. 1 simben90 cis90   3766 Sep 12 18:53 mbox
drwxr-xr-x. 2 simben90 cis90   4096 Oct  5 10:30 misc
drwxr-xr-x. 7 simben90 cis90   4096 Oct  5 10:35 poems
-r--------. 1 simben90 staff   5899 Oct  4 11:04 test01.graded
-rw-rw-r--. 1 simben90 cis90  17341 Sep 19 19:31 uhistory
/home/cis90/simben $
```

This column shows the username that owns the file

File Permissions

Interpreting the permission codes on long listings

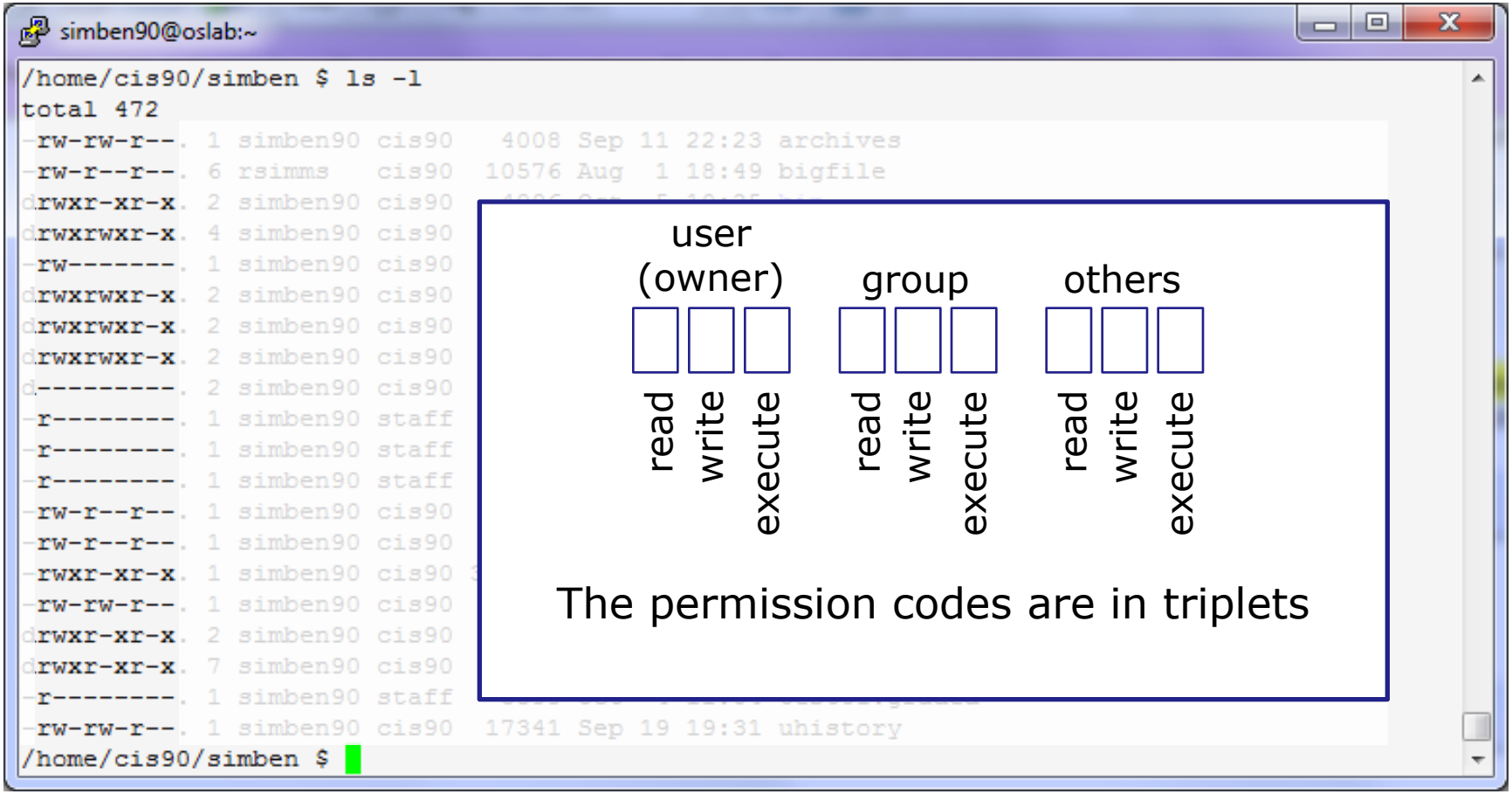
simben90@oslab:~

```
/home/cis90/simben $ ls -l
total 472
-rw-rw-r--. 1 simben90 cis90   4008 Sep 11 22:23 archives
-rw-r--r--. 6 rsimms   cis90  10576 Aug  1 18:49 bigfile
drwxr-xr-x. 2 simben90 cis90   4096 Oct  5 10:25 bin
drwxrwxr-x. 4 simben90 cis90   4096 Oct  5 10:21 class
-rw-------. 1 simben90 cis90   1894 Sep 20 06:23 dead.letter
drwxrwxr-x. 2 simben90 cis90   4096 Oct  5 10:25 docs
drwxrwxr-x. 2 simben90 cis90   4096 Oct  5 10:30 edits
drwxrwxr-x. 2 simben90 cis90   4096 Oct  5 10:41 etc
d---------. 2 simben90 cis90   4096 Feb  1  2002 Hidden
-r--------. 1 simben90 staff   2780 Sep  6 13:47 lab01.graded
-r--------. 1 simben90 staff   1312 Sep 13 12:27 lab02.graded
-r--------. 1 simben90 staff    814 Sep 27 13:08 lab04.graded
-rw-r--r--. 1 simben90 cis90   1059 Oct  7 14:41 letter
-rw-r--r--. 1 simben90 cis90    208 Oct  5 10:45 log
-rwxr-xr-x. 1 simben90 cis90 375252 Oct  7 14:05 mail
-rw-rw-r--. 1 simben90 cis90   3766 Sep 12 18:53 mbox
drwxr-xr-x. 2 simben90 cis90   4096 Oct  5 10:30 misc
drwxr-xr-x. 7 simben90 cis90   4096 Oct  5 10:35 poems
-r--------. 1 simben90 staff   5899 Oct  4 11:04 test01.graded
-rw-rw-r--. 1 simben90 cis90  17341 Sep 19 19:31 uhistory
/home/cis90/simben $
```

This column shows the group each file belong to

File Permissions

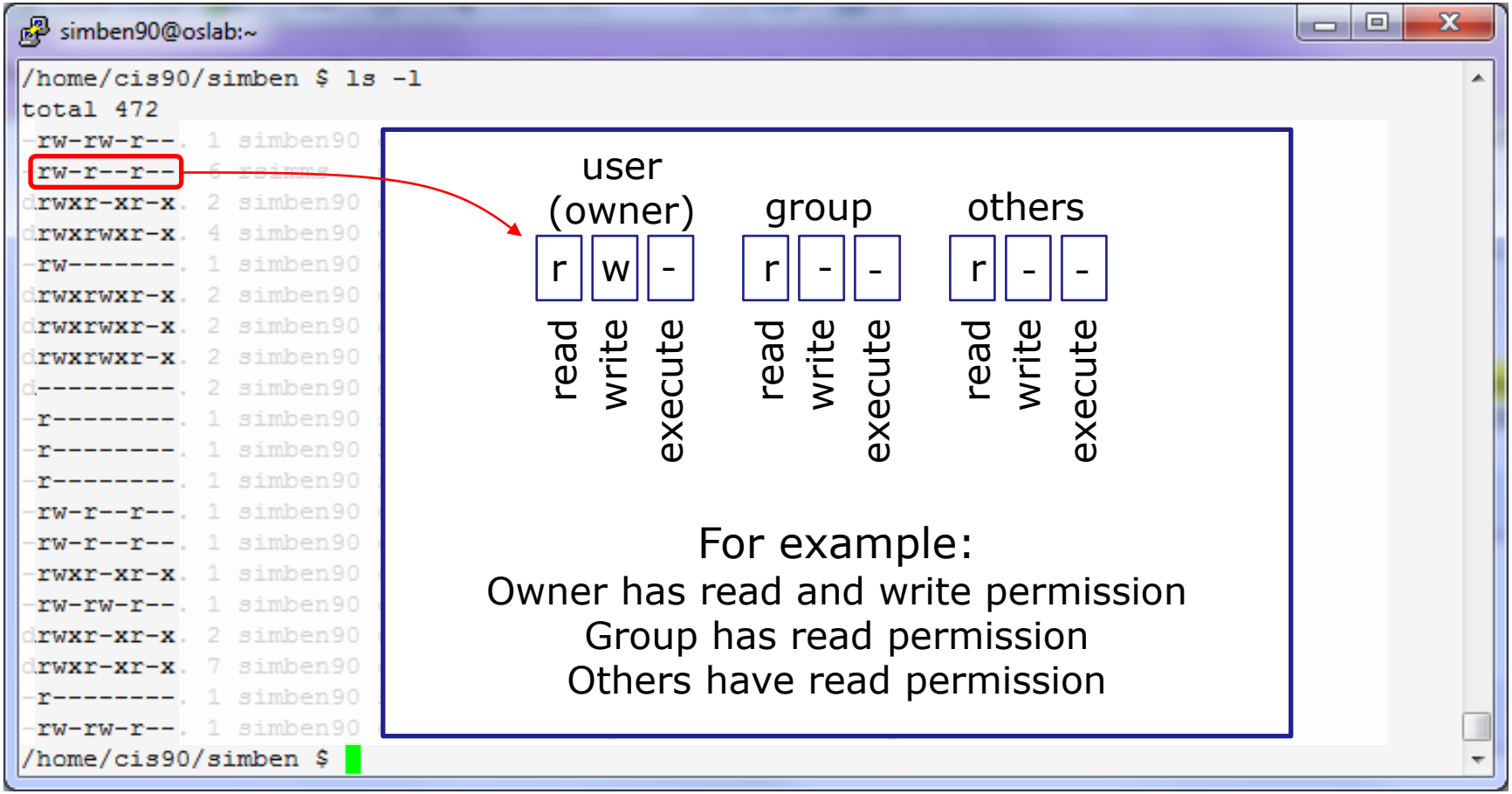
Interpreting the permission codes on long listings



The 9 permission bits are grouped by ***user*** *(owner),* ***group*** *and all* ***others***

File Permissions

Interpreting the permission codes on long listings



*Individual permission settings can be set for the **user** (owner), **group** and all **others***


File Permissions
Read

Read permission is necessary to read a file

```
/home/cis90/simben $ ls -l /etc/passwd /etc/shadow
-rw-r--r--. 1 root root  7990 Oct  4 08:02 /etc/passwd
----------. 1 root root 11944 Oct  3 11:48 /etc/shadow
```

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

Benji, considered as "other", has read permission to /etc/passwd

```
/home/cis90/simben $ 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/cis90/simben $ head -3 /etc/shadow
head: cannot open `/etc/shadow' for reading: Permission denied
```


File Permissions
Write


Write permission is necessary to write to a file

```
/home/cis90/simben $ ls -l letter ../rodduk/letter
-rw-r--r--. 1 simben90 cis90 1059 Oct  7 15:05 letter
-rw-r--r--. 1 rodduk90 cis90 1044 Jul 20  2001 ../rodduk/letter
```

These files have different owners but are in the same group

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

```
/home/cis90/simben $ echo "Benji was here" >> 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
```


File Permissions
Execute


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

```
/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
-rwxrw---+. 1 rsimms   staff 8718 Aug  1 18:37 ../bin/check7
-rwxr-xr-x. 1 simben90 cis90  174 Mar  4  2004 bin/tryme
```

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
```

Groups and new files

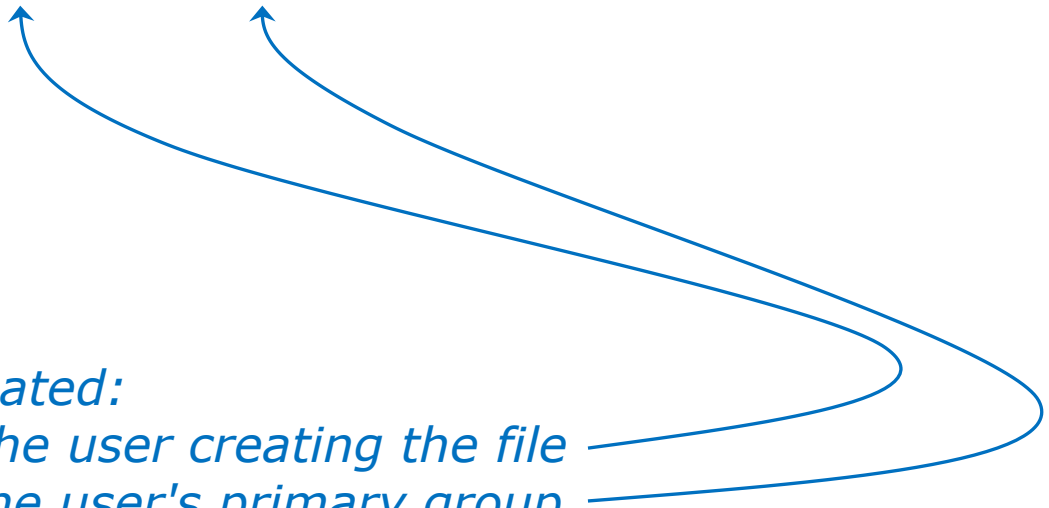
More tools for your toolbox

groups – displays file inode information (status) and more

id – displays information about a user

Groups

```
/home/cis90/simben $ touch mydogs
/home/cis90/simben $ ls -l mydogs
-rw-rw-r--. 1 simben90 cis90 0 Oct  7 15:12 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*

```
/home/cis90/simben $ id simben90
uid=1001(simben90) gid=190(cis90)
groups=190(cis90),100(users)

/home/cis90/simben $ groups simben90
simben90 : cis90 users
```

Primary group (cis90)

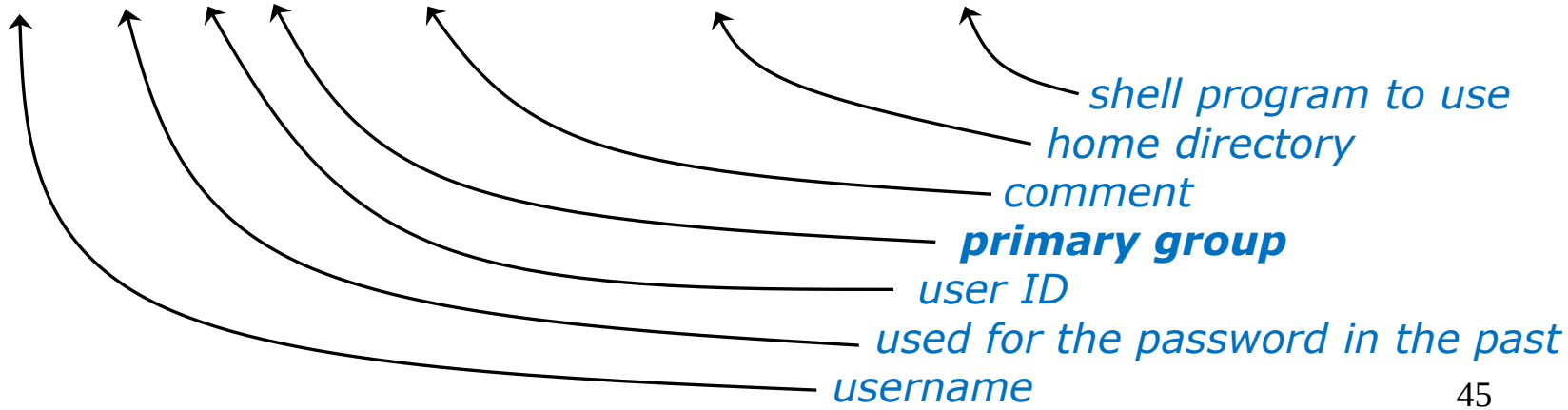
Secondary group (users)

Primary group recorded in /etc/passwd

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

Excerpt from **/etc/passwd**

```
cis90:x:1000:190:CIS 90 Student:/home/cis90/cis:/bin/bash
simben90:x:1001:190:Benji Simms:/home/cis90/simben:/bin/bash
milhom90:x:1002:190:Homer Miller:/home/cis90/milhom:/bin/bash
rodduk90:x:1003:190:Duke Roddy:/home/cis90/rodduk:/bin/bash
calsea90:x:1006:190:Sean Callahan:/home/cis90/calsea:/bin/bash
davdon90:x:1007:190:Don Davis:/home/cis90/davdon:/bin/bash
ellcar90:x:1008:190:Carlile Ellis:/home/cis90/ellcar:/bin/bash
frocar90:x:1009:190:Carter Frost:/home/cis90/frocar:/bin/bash
hendaj90:x:1010:190:Dajan Henk:/home/cis90/hendaj:/bin/bash
kanbry90:x:1011:190:Bryn Kanar:/home/cis90/kanbry:/bin/bash
kenrit90:x:1012:190:Rita Kennedy:/home/cis90/kenrit:/bin/bash
```



Secondary groups recorded in /etc/group

Secondary group membership is recorded in /etc/group

Excerpts from **/etc/group**

```
audio:x:63:
nobody:x:99:
users:x:100:guest,jimg,rsimms,gerlinde,cis90,simben90,milhom90,rodduk90,calsea90,davd
on90,ellcar90,frocar90,hendaj90,kanbry90,kenrit90,libkel90,lyoben90,marray90,menfid90
,mesmic90,noreva90,potjos90,ramgus90,wiljac90,zamhum90,fyosea90,verevi90,rawjes90,mes
cha90,evaand90,ahrmat98,calsea98,capchr98,colabd98,dinchr98,doucor98,drybry98,flamat9
8,goothe98,lewzar98,mccmic98,roclea98,shidev98,sonely98,srelau98,syljos98,thepat98,va
rana98,veleli98,wildan98,alvdes98,musdav98,luztas98,visgab98,fareli98,ramcar90,chiand
98,farsha90,arcmat172,balcor172,bodian172,deddil172,dusaar172,evaand172,sha172,galgwy
172,gilgab172,hilsco172,juarub172,mic172,lemrya172,maradr172,matmar172,melale172,menf
id172,monlui172,mordav172,pallar172,perste172,rodchr172,rutsam172,schjon172,weltod172
,wiltyr172,wismar172,bramar172,172,acctes172,bermic172,lejmic172,farsha172,ianbod172
dbus:x:81:
utmp:x:22:
< snipped >
guest:x:506:
staff:x:503:rsimms,gerlinde,jimg,rick
cis90:x:190:guest,rsimms,jimg
cis98:x:130:jimg,rsimms
cis172:x:172:gerlinde
cis191:x:191:rsimms,jimg
cis192:x:192:rsimms,jimg
```

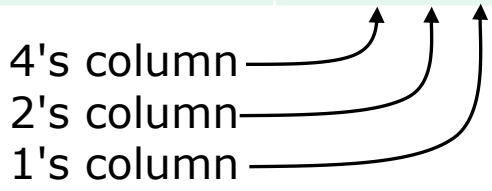
Specifying Numerical Permissions

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



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)

= 110 or 4+2+0 = 6

binary *decimal* *decimal*

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)

= 011 *binary* or 0+2+1 *decimal* = 3 *decimal*

Practice converting to numerical

File Permissions

Interpreting the permission codes on long listings

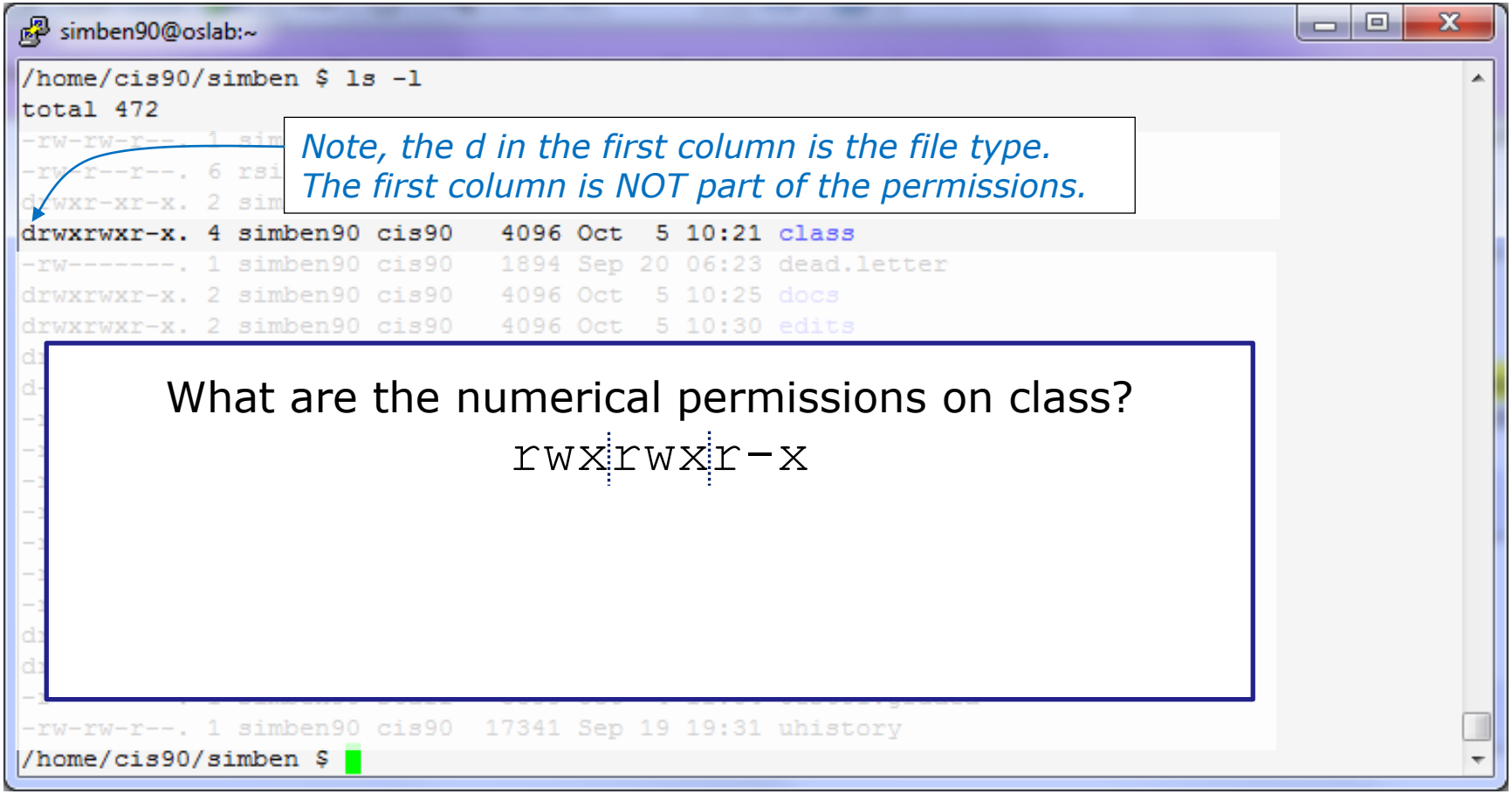
simben90@oslab:~

```
/home/cis90/simben $ ls -l
total 472
-rw-rw-r--. 1 simben90 cis90   4008 Sep 11 22:23 archives
-rw-r--r--. 6 rsimms   cis90  10576 Aug  1 18:49 bigfile
drwxr-xr-x. 2 simben90 cis90   4096 Oct  5 10:25 bin
drwxrwxr-x. 4 simben90 cis90   4096 Oct  5 10:21 class
-rw-------. 1 simben90 cis90   1894 Sep 20 06:23 dead.letter
drwxrwxr-x. 2 simben90 cis90   4096 Oct  5 10:25 docs
drwxrwxr-x. 2 simben90 cis90   4096 Oct  5 10:30 edits
drwxrwxr-x. 2 simben90 cis90   4096 Oct  5 10:41 etc
d---------. 2 simben90 cis90   4096 Feb  1  2002 Hidden
-r--------. 1 simben90 staff   2780 Sep  6 13:47 lab01.graded
-r--------. 1 simben90 staff   1312 Sep 13 12:27 lab02.graded
-r--------. 1 simben90 staff    814 Sep 27 13:08 lab04.graded
-rw-r--r--. 1 simben90 cis90   1059 Oct  7 14:41 letter
-rw-r--r--. 1 simben90 cis90    208 Oct  5 10:45 log
-rwxr-xr-x. 1 simben90 cis90 375252 Oct  7 14:05 mail
-rw-rw-r--. 1 simben90 cis90   3766 Sep 12 18:53 mbox
drwxr-xr-x. 2 simben90 cis90   4096 Oct  5 10:30 misc
drwxr-xr-x. 7 simben90 cis90   4096 Oct  5 10:35 poems
-r--------. 1 simben90 staff   5899 Oct  4 11:04 test01.graded
-rw-rw-r--. 1 simben90 cis90  17341 Sep 19 19:31 uhistory
/home/cis90/simben $
```

This is a long listing of Benji's home directory

Example 1

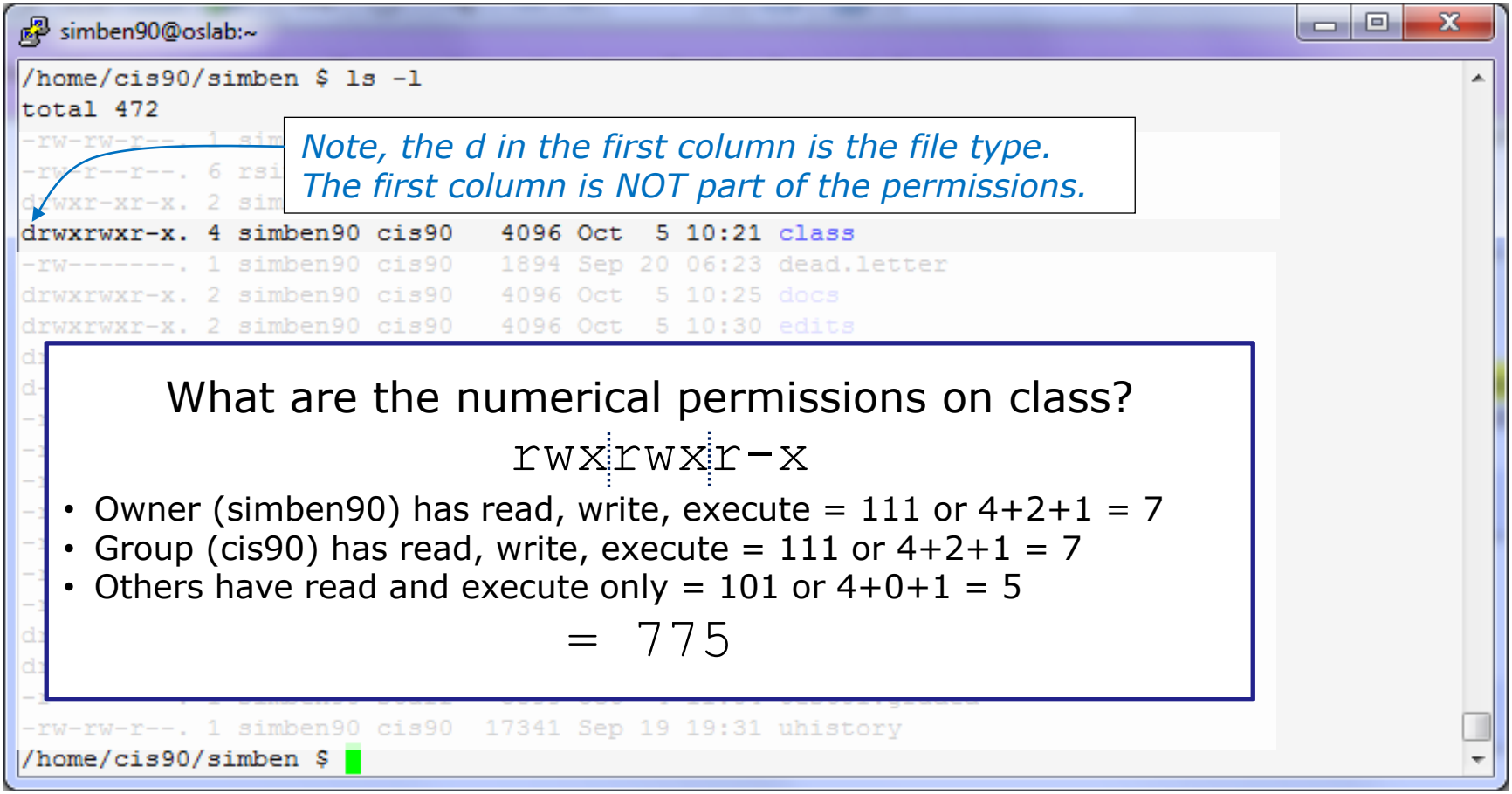
Converting mnemonic permissions to numeric



Benji's class (directory)

Example 1

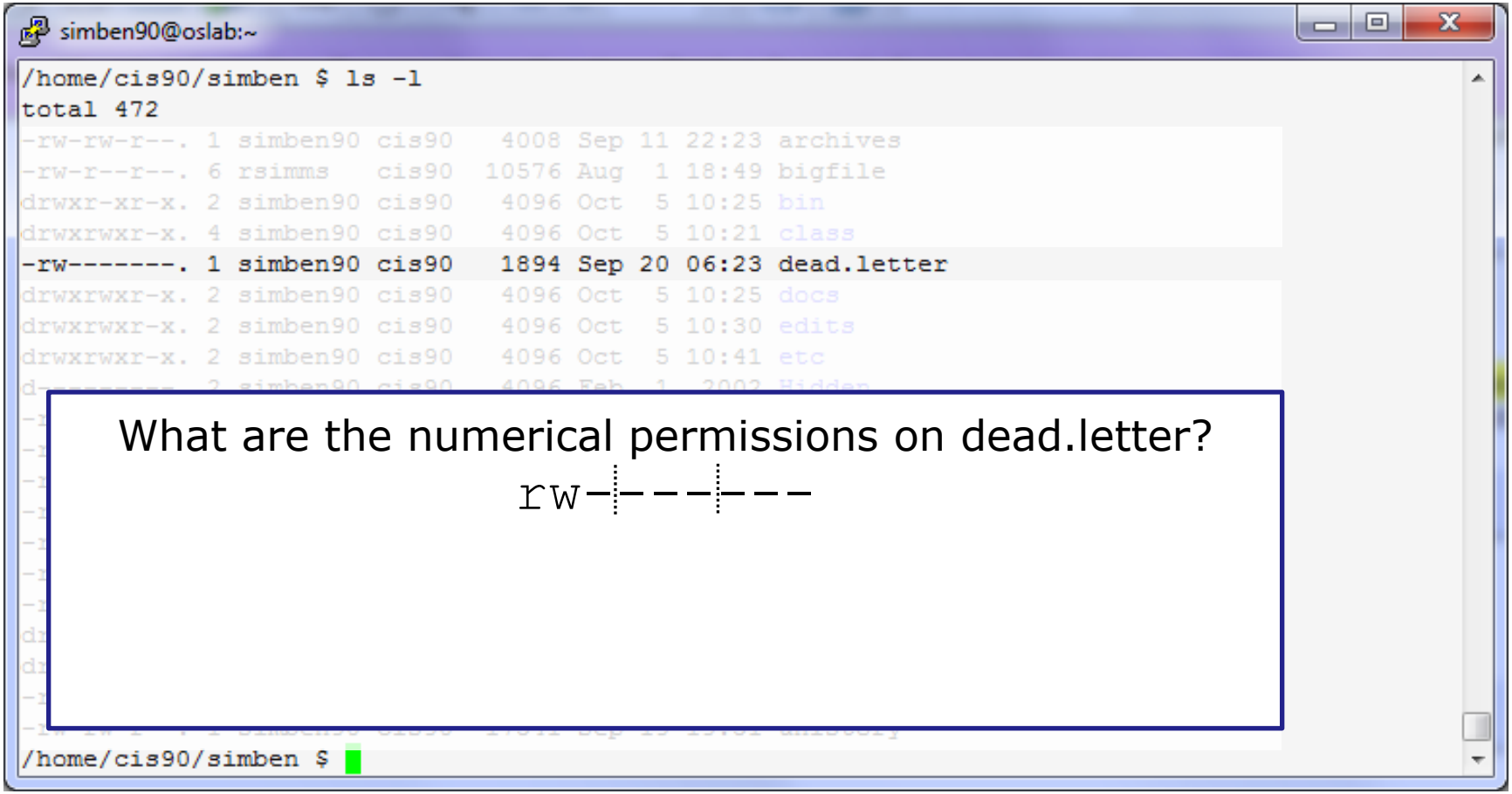
Converting mnemonic permissions to numeric



Benji's class (directory)

Example 2

Converting mnemonic permissions to numeric



Benji's dead.letter (regular file)

Example 2

Converting mnemonic permissions to numeric

simben90@oslab:~

```
/home/cis90/simben $ ls -l
total 472
-rw-rw-r--. 1 simben90 cis90   4008 Sep 11 22:23 archives
-rw-r--r--. 6 rsimms   cis90  10576 Aug  1 18:49 bigfile
drwxr-xr-x. 2 simben90 cis90   4096 Oct  5 10:25 bin
drwxrwxr-x. 4 simben90 cis90   4096 Oct  5 10:21 class
-rw-------. 1 simben90 cis90   1894 Sep 20 06:23 dead.letter
drwxrwxr-x. 2 simben90 cis90   4096 Oct  5 10:25 docs
drwxrwxr-x. 2 simben90 cis90   4096 Oct  5 10:30 edits
drwxrwxr-x. 2 simben90 cis90   4096 Oct  5 10:41 etc
/home/cis90/simben $
```

What are the numerical permissions on dead.letter?

rw-|---|---

- Owner (simben90) has read and write only = 110 or 4+2+0 = 6
- Group (cis90) has no permissions = 000 or 0+0+0 = 0
- Others have no permission = 000 or 0+0+0 = 0

= 600

Benji's dead.letter (regular file)

Example 3

Converting mnemonic permissions to numeric

simben90@oslab:~

```
/home/cis90/simben $ ls -l
total 472
-rw-rw-r--. 1 simben90 cis90   4008 Sep 11 22:23 archives
-rw-r--r--. 6 rsimms   cis90  10576 Aug  1 18:49 bigfile
drwxr-xr-x. 2 simben90 cis90   4096 Oct  5 10:25 bin
drwx
-rw-
drwx
drwx
drwx
d---
-r--
-r--
-r--
-rw-
-rw-
-rwx
-rw-rw-r--. 1 simben90 cis90   3766 Sep 12 18:53 mbox
drwxr-xr-x. 2 simben90 cis90   4096 Oct  5 10:30 misc
drwxr-xr-x. 7 simben90 cis90   4096 Oct  5 10:35 poems
-r--------. 1 simben90 staff   5899 Oct  4 11:04 test01.graded
-rw-rw-r--. 1 simben90 cis90  17341 Sep 19 19:31 uhistory
/home/cis90/simben $
```

What are the numerical permissions on test01.graded?

r--------

Benji's test01.graded (regular file)

Example 3

Converting mnemonic permissions to numeric

simben90@oslab:~

```
/home/cis90/simben $ ls -l
total 472
-rw-rw-r--. 1 simben90 cis90   4008 Sep 11 22:23 archives
-rw-r--r--. 6 rsimms   cis90  10576 Aug  1 18:49 bigfile
drwxr-xr-x. 2 simben90 cis90   4096 Oct  5 10:25 bin
drwx
-rw-
drwx
drwx
drwx
d---
-r--
-r--
-r--
-rw-
-rw-
-rwx
-rw-rw-r--. 1 simben90 cis90   3766 Sep 12 18:53 mbox
drwxr-xr-x. 2 simben90 cis90   4096 Oct  5 10:30 misc
drwxr-xr-x. 7 simben90 cis90   4096 Oct  5 10:35 poems
-r--------. 1 simben90 staff   5899 Oct  4 11:04 test01.graded
-rw-rw-r--. 1 simben90 cis90  17341 Sep 19 19:31 uhistory
/home/cis90/simben $
```

What are the numerical permissions on test01.graded?

r--|---|---

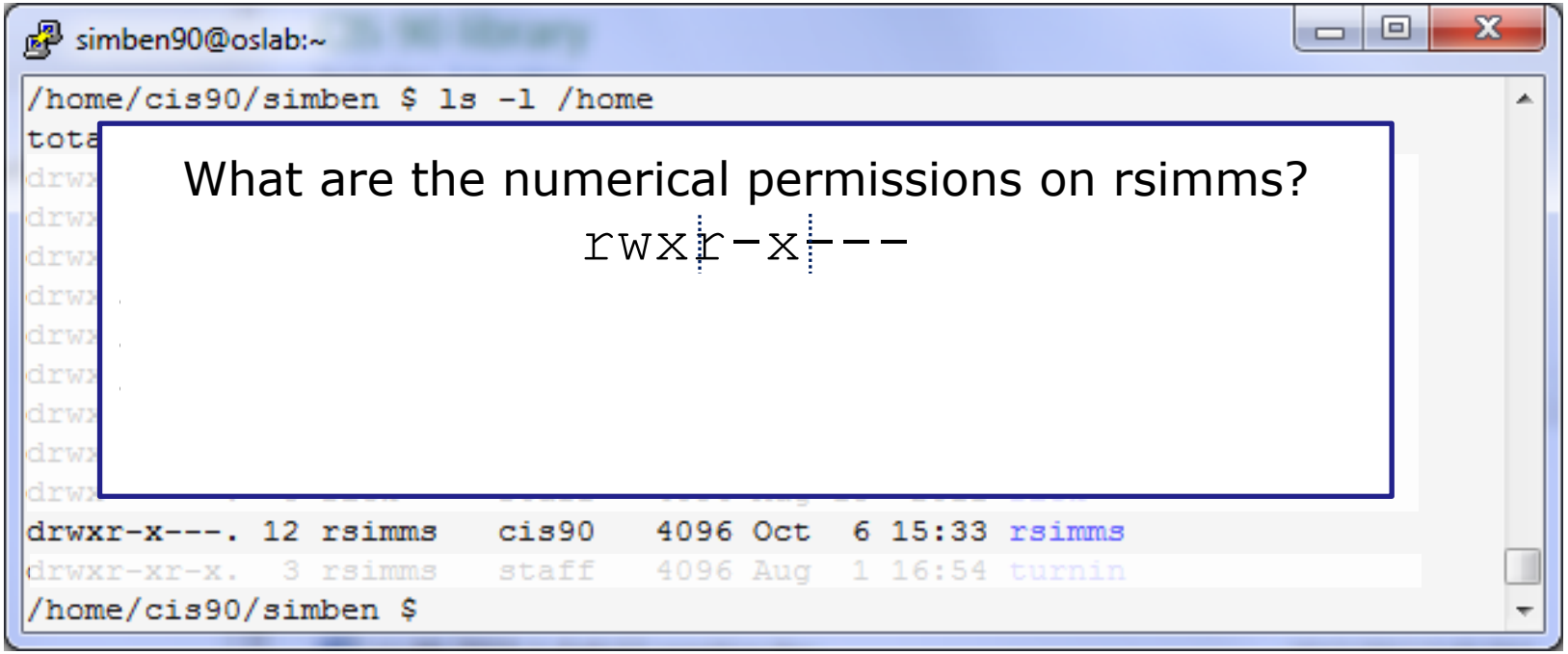
- Owner (simben90) has read only = 100 or 4+0+0 = 4
- Group (staff) has no permissions = 000 or 0+0+0 = 0
- Others have no permission = 000 or 0+0+0 = 0

= 400

Benji's test01.graded (regular file)

Example 4

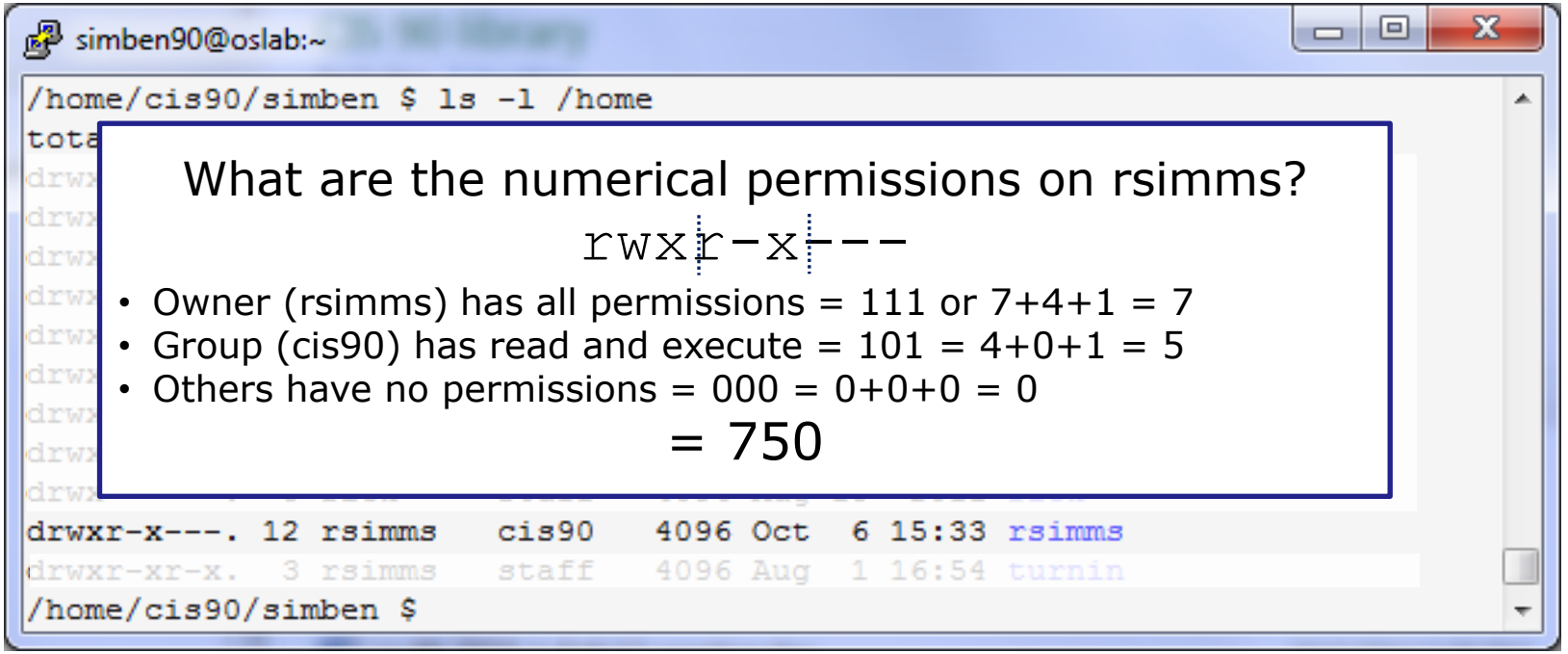
Converting mnemonic permissions to numeric



/home/rsimms (Rich's home directory)

Example 4

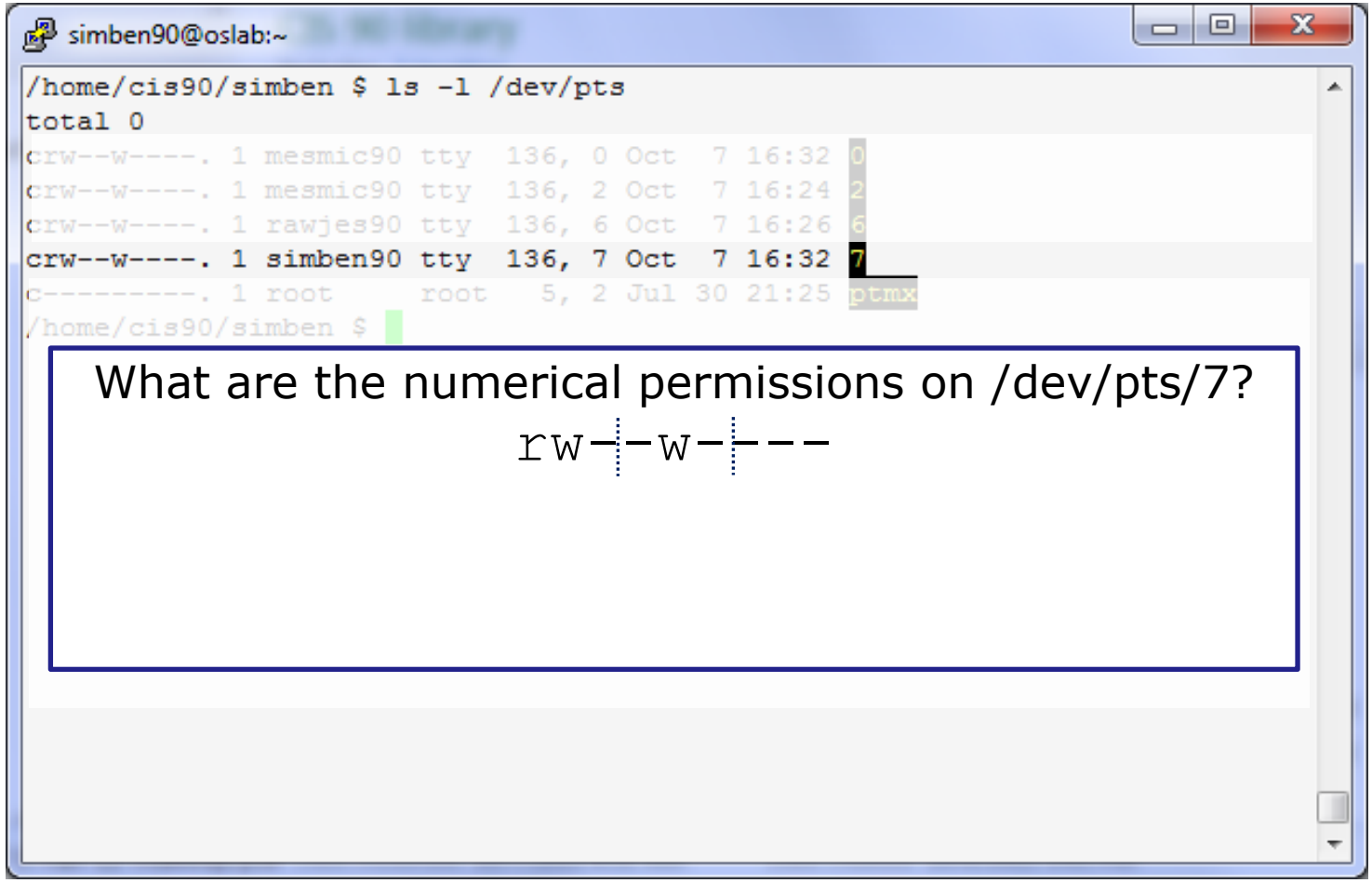
Converting mnemonic permissions to numeric



/home/rsimms (Rich's home directory)

Example 5

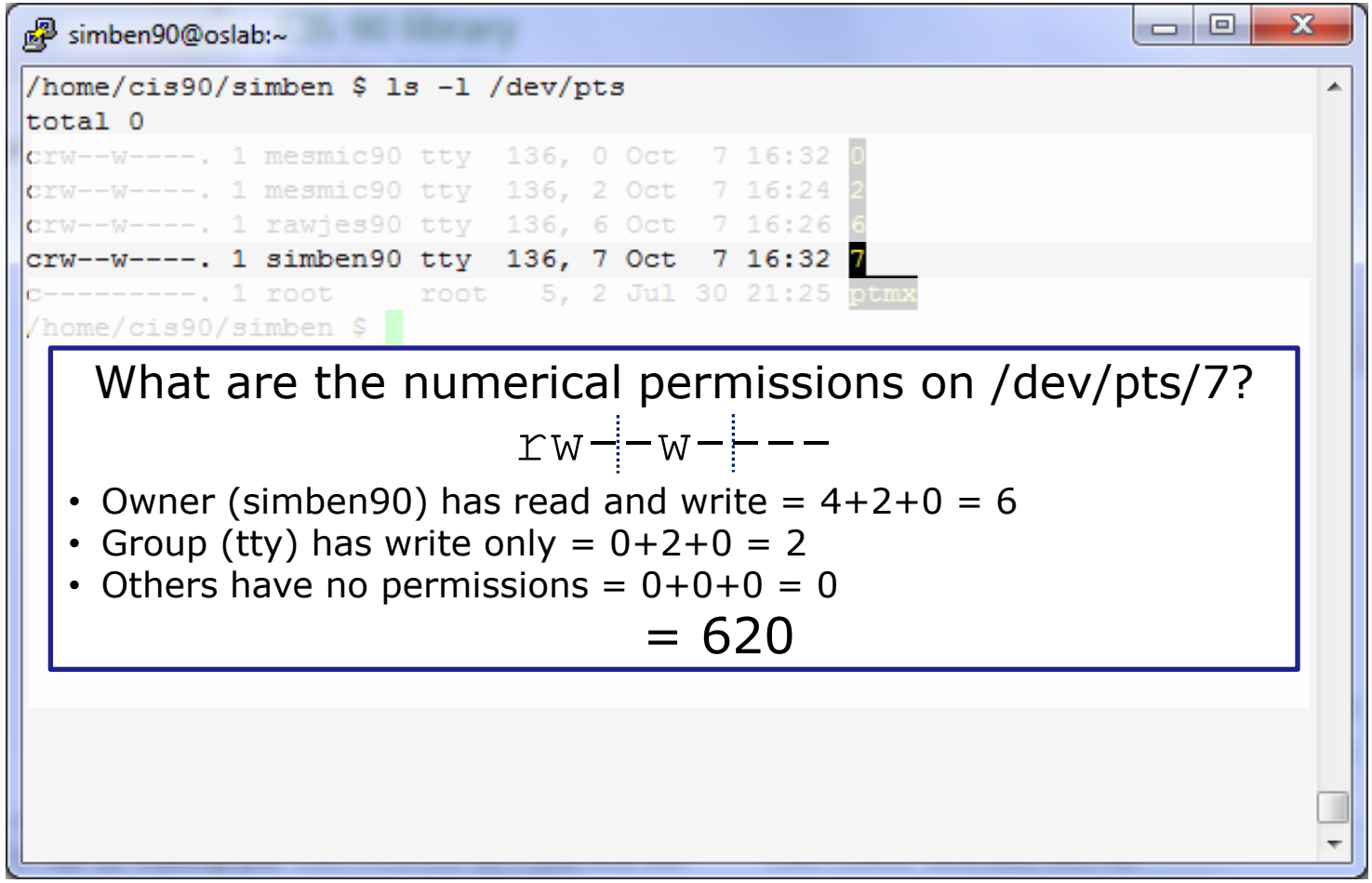
Converting mnemonic permissions to numeric



/dev/pts/7 (character special device file)

Example 5

Converting mnemonic permissions to numeric



/dev/pts/7 (character special device file)

Recap

File Permissions

Summary

How do we control access to files and directories?

How do we control access to files and directories?

Answer: **file permissions**

File Permissions
Summary

What permissions are there?

File Permissions
Summary

What permissions are there?

Answer: **read, write and execute**

File Permissions
Summary

Who do permissions apply to?

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**)

Letter file in detail

Tools for your toolbox

ls -l – produces a "long listing" showing some of the inode information

stat – file "status" which displays additional inode information and more

File Permissions

Relevant fields from the inode

```
/home/cis90/simmsben $ ls -l
total 176
total 472
-rw-rw-r--. 1 simben90 cis90   4008 Sep 11 22:23 archives
-rw-r--r--. 6 rsimms   cis90  10576 Aug  1 18:49 bigfile
drwxr-xr-x. 2 simben90 cis90   4096 Oct  5 10:25 bin
drwxrwxr-x. 4 simben90 cis90   4096 Oct  5 10:21 class
-rw-------. 1 simben90 cis90   1894 Sep 20 06:23 dead.letter
drwxrwxr-x. 2 simben90 cis90   4096 Oct  5 10:25 docs
drwxrwxr-x. 2 simben90 cis90   4096 Oct  5 10:30 edits
drwxrwxr-x. 2 simben90 cis90   4096 Oct  5 10:41 etc
d---------. 2 simben90 cis90   4096 Feb  1  2002 Hidden
-r--------. 1 simben90 staff   2780 Sep  6 13:47 lab01.graded
-r--------. 1 simben90 staff   1312 Sep 13 12:27 lab02.graded
-r--------. 1 simben90 staff    814 Sep 27 13:08 lab04.graded
-rw-r--r--. 1 simben90 cis90   1059 Oct  7 15:05 letter
-rw-r--r--. 1 simben90 cis90    208 Oct  5 10:45 log
-rwxr-xr-x. 1 simben90 cis90 375252 Oct  7 14:05 mail
-rw-rw-r--. 1 simben90 cis90   3766 Sep 12 18:53 mbox
drwxr-xr-x. 2 simben90 cis90   4096 Oct  5 10:30 misc
-rw-rw-r--. 1 simben90 cis90      0 Oct  7 15:12 mydogs
drwxr-xr-x. 7 simben90 cis90   4096 Oct  5 10:35 poems
-r--------. 1 simben90 staff   5899 Oct  4 11:04 test01.graded
-rw-rw-r--. 1 simben90 cis90  17341 Sep 19 19:31 uhistory
```

Permissions *Owner* *Group*

FYI:
In newer distros, GNU ls uses a '.' character to indicate a file with an SELinux security context, but no other alternate access method.

http://www.gnu.org/software/coreutils/manual/html_node/What-information-is-listed.html#What-information-is-listed

File Permissions

Relevant fields from the inode

```
/home/cis90/simmsben $ ls -l
total 176
total 472
-rw-rw-r--. 1 simben90 cis90   4008 Sep 11 22:23 archives
-rw-r--r--. 6 rsimms   cis90  10576 Aug  1 18:49 bigfile
drwxr-xr-x. 2 simben90 cis90   4096 Oct  5 10:25 bin
drwxrwxr-x. 4 simben90 cis90   4096 Oct  5 10:21 class
-rw-------. 1 simben90 cis90   1894 Sep 20 06:23 dead.letter
drwxrwxr-x. 2 simben90 cis90   4096 Oct  5 10:25 docs
drwxrwxr-x. 2 simben90 cis90   4096 Oct  5 10:30 edits
drwxrwxr-x. 2 simben90 cis90   4096 Oct  5 10:41 etc
d---------. 2 simben90 cis90   4096 Feb  1  2002 Hidden
-r--------. 1 simben90 staff   2780 Sep  6 13:47 lab01.graded
-r--------. 1 simben90 staff   1312 Sep 13 12:27 lab02.graded
-r--------. 1 simben90 staff    814 Sep 27 13:08 lab04.graded
-rw-r--r--. 1 simben90 cis90   1059 Oct  7 15:05 letter
-rw-r--r--. 1 simben90 cis90    208 Oct  5 10:45 log
-rwxr-xr-x. 1 simben90 cis90 375252 Oct  7 14:05 mail
-rw-rw-r--. 1 simben90 cis90   3766 Sep 12 18:53 mbox
drwxr-xr-x. 2 simben90 cis90   4096 Oct  5 10:30 misc
-rw-rw-r--. 1 simben90 cis90      0 Oct  7 15:12 mydogs
drwxr-xr-x. 7 simben90 cis90   4096 Oct  5 10:35 poems
-r--------. 1 simben90 staff   5899 Oct  4 11:04 test01.graded
-rw-rw-r--. 1 simben90 cis90  17341 Sep 19 19:31 uhistory
```

Permissions *Owner* *Group*

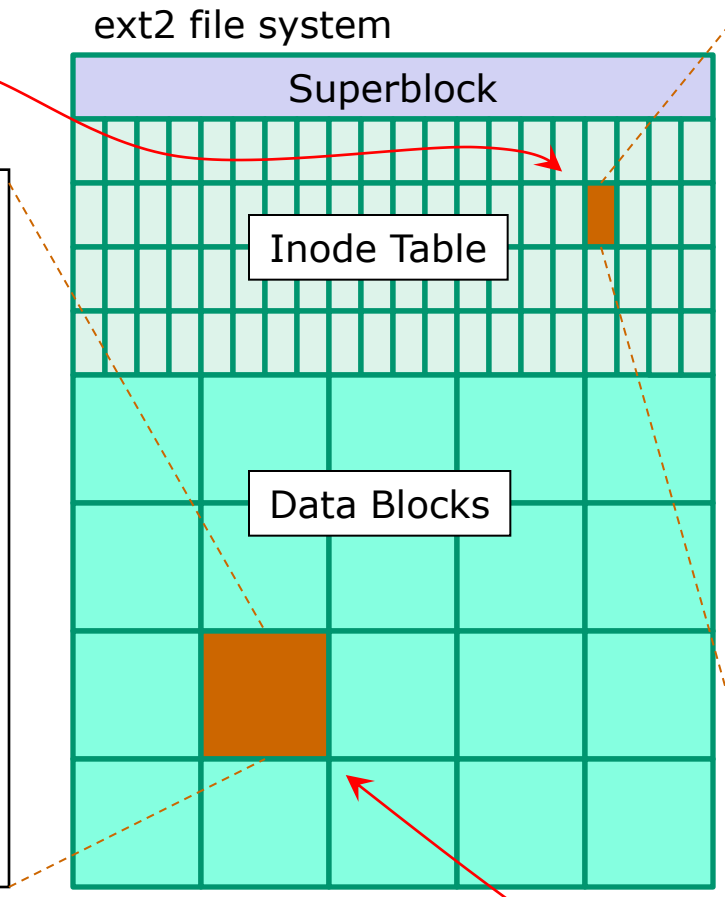
The owner of letter is simben90 and the group is cis90

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

The filename is kept in the directory

bigfile 12687
bin 12067
letter 10574

Permissions, owner, group, etc. are kept in the inode



10574	inode number
-	Type
rw-r—r--	Permissions
1	Number of links
simben90	User
cis90	Group
1059	Size
2012-10-07	Modification time
2012-10-07	Access Time
2012-10-07	Change time
Pointer(s) to data blocks	Pointer(s) to data blocks

Hello Mother! Hello Father!

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

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

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

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

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

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

Alan Sherman

The actual content is kept in a data block

```
/home/cis90/simmsben $ ls -il letter
10574 -rw-r--r--. 1 simben90 cis90  1059 Oct  7 15:05 letter
```

File Permissions

Example: letter file

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

```
/home/cis90/simben $ stat letter
  File: `letter'
  Size: 1059            Blocks: 8          IO Block: 4096
     regular file
Device: 805h/2053d      Inode: 10574       Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1001/simben90)   Gid: (  190/
     cis90)
Access: 2012-10-07 15:06:09.922703386 -0700
Modify: 2012-10-07 15:05:57.856733896 -0700
Change: 2012-10-07 15:05:57.856733896 -0700
/home/cis90/simben $
```

numeric form

110100100

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

owner has read and write

group has only read

others have only read

More Practice

File Permissions

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

File Permissions

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

4 4 0

`100100000`

Answer: 440

Owner has read
Group has read
Others have no permissions

File Permissions

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

File Permissions

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

7 6 4

111 110 100

Answer: 764

File Permissions

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

File Permissions

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

7 5 5

111 101 101

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?

```
110 100 100
rw- 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 9778 Apr 30 11:35 /etc/samba/smb.conf
```

root has read & write
root group has read
all other users, including simben90, have read

Configuring Permissions

Tools for your toolbox

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 file mode "permission" bits of a file.

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

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

umask – Allows you to fully control the permissions new files and directories are created with

chown

chown – change owner

Syntax:

chown *newowner pathname(s)*

Examples:

- **`chown rsimms letter`**
- **`chown simben90 lab*.graded`**
- **`chown rsimms /home/cis90/bin/*`**

chown – change owner

```
/home/cis90/milhom $ touch myfile
/home/cis90/milhom $ ls -l myfile
-rw-rw-r--. 1 milhom90 cis90 0 Oct  9 10:23 myfile

/home/cis90/milhom $ chown simben90 myfile
chown: changing ownership of `myfile': Operation not permitted
```

Make a test file and try to change the owner

*Only root can use the **chown** command*

```
/home/cis90/milhom $ su -
Password:
[root@oslab ~]# chown simben90 /home/cis90/milhom/myfile
[root@oslab ~]# ls -l /home/cis90/milhom/myfile
-rw-rw-r--. 1 simben90 cis90 0 Oct  9 10:23 /home/cis90/milhom/myfile
[root@oslab ~]#
```

chgrp

chgrp – change group

Syntax:

chgrp *group pathname(s)*

Examples:

- **`chgrp users letter`**
- **`chgrp cis90 /home/cis90/bin/*`**

chgrp – change group

```
/home/cis90/milhom $ ls -l myfile
-rw-rw-r--. 1 milhom90 cis90 0 Oct  9 10:23 myfile
```

change group to users

```
/home/cis90/milhom $ chgrp users myfile
/home/cis90/milhom $ ls -l myfile
-rw-rw-r--. 1 milhom90 users 0 Oct  9 10:23 myfile
```

change group back to cis90

```
/home/cis90/milhom $ chgrp cis90 myfile
/home/cis90/milhom $ ls -l myfile
-rw-rw-r--. 1 milhom90 cis90 0 Oct  9 10:23 myfile
```

You can only change the group on a file or directory t one that you belong to

chmod

chmod – change permissions

Syntax:

chmod permissions *pathname(s)*

may be specified numerically or mnemonically

Examples:

- **chmod 750 check5 check6**
- **chmod 644 poems/*/***

numeric

- **chmod +x myscript**
- **chmod g+rw share/***

mnemonic

chmod (mnemonic)

Mnemonic permission specifications

Relative changes to the files previous permissions

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 the commas!*

chmod using mnemonics

mnemonics – "memory aids"

The file does not currently have execute permission for the user or group

```
/home/cis90/milhom $ ls -l myfile
-rw-rw-r--. 1 milhom90 cis90 0 Oct  9 10:23 myfile
```

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

```
/home/cis90/milhom $ chmod u+x myfile
/home/cis90/milhom $ ls -l myfile
-rwxrw-r--. 1 milhom90 cis90 0 Oct  9 10:23 myfile
```

add execute permission for user (owner)

```
/home/cis90/milhom $ chmod g+x myfile
/home/cis90/milhom $ ls -l myfile
-rwxrwxr--. 1 milhom90 cis90 0 Oct  9 10:23 myfile
```

add execute permission for group

Use chmod to add or remove permissions from a file

chmod using mnemonics

mnemonics – "memory aids"

```
/home/cis90/milhom $ ls -l myfile
-rwxrwxr--. 1 milhom90 cis90 0 Oct  9 10:23 myfile

/home/cis90/milhom $ chmod -x myfile
/home/cis90/milhom $ ls -l myfile
-rw-rw-r--. 1 milhom90 cis90 0 Oct  9 10:23 myfile
/home/cis90/milhom $ chmod go+x myfile

/home/cis90/milhom $ ls -l myfile
-rw-rwxr-x. 1 milhom90 cis90 0 Oct  9 10:23 myfile
/home/cis90/milhom $ chmod go-rwx myfile

/home/cis90/milhom $ ls -l myfile
-rw-------. 1 milhom90 cis90 0 Oct  9 10:23 myfile
```

remove execute from all

add execute to others and group

remove read, write, execute from groups and others

Use chmod to add or remove permissions from a file

chmod (numerical)

chmod using numerical method

```
/home/cis90/milhom $ ls -l myfile
-rw-------. 1 milhom90 cis90 0 Oct  9 10:23 myfile

/home/cis90/milhom $ chmod 664 myfile
/home/cis90/milhom $ ls -l myfile
-rw-rw-r--. 1 milhom90 cis90 0 Oct  9 10:23 myfile
```

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

chmod using numerical method

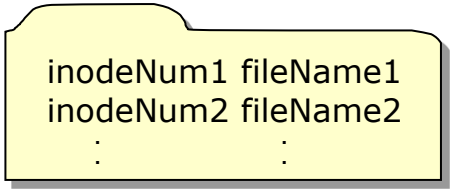
```
/home/cis90/milhom $ chmod 777 myfile
/home/cis90/milhom $ ls -l myfile
-rwxrwxrwx. 1 milhom90 cis90 0 Oct  9 10:23 myfile

/home/cis90/milhom $ chmod 640 myfile
/home/cis90/milhom $ ls -l myfile
-rw-r-----. 1 milhom90 cis90 0 Oct  9 10:23 myfile
/home/cis90/milhom $ chmod 000 myfile

/home/cis90/milhom $ ls -l myfile
----------. 1 milhom90 cis90 0 Oct  9 10:23 myfile
/home/cis90/milhom $ chmod 644 myfile

/home/cis90/milhom $ ls -l myfile
-rw-r--r--. 1 milhom90 cis90 0 Oct  9 10:23 myfile
```

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

File Permissions in action

File Permissions

Commands that use file permissions



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:

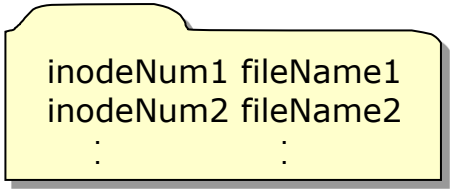
```
/home/cis90/simmsben/Directory3 $ echo Blah Blah Blah > myfile
/home/cis90/simmsben/Directory3 $ cat myfile
Blah Blah Blah
/home/cis90/simmsben/Directory3 $ chmod u-r myfile
/home/cis90/simmsben/Directory3 $ ls -l myfile
--w-r--r-- 1 simmsben cis90 15 Oct 13 08:50 myfile
/home/cis90/simmsben/Directory3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



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 :

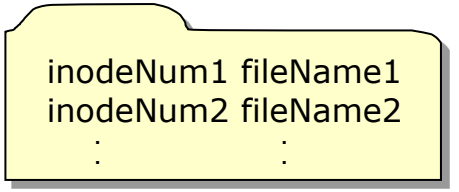
```
/home/cis90/simmsben/Directory3 $ echo Blah Blah Blah > myfile
/home/cis90/simmsben/Directory3 $ chmod 444 myfile
/home/cis90/simmsben/Directory3 $ ls -l myfile
-r--r--r-- 1 simmsben cis90 15 Oct 13 09:02 myfile
/home/cis90/simmsben/Directory3 $ echo Blah Blah Blah > myfile
-bash: myfile: Permission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

write permission removed

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

File Permissions

Commands that use file permissions



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:

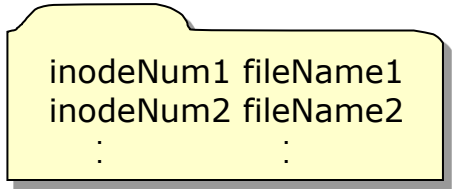
```
/home/cis90/simmsben/Directory3 $ echo "banner $LOGNAME is cool" > myfile
/home/cis90/simmsben/Directory3 $ cat myfile
banner $LOGNAME is cool
/home/cis90/simmsben/Directory3 $ myfile
-bash: ./myfile: Permission denied
/home/cis90/simmsben/Directory3 $ chmod +x myfile
/home/cis90/simmsben/Directory3 $ 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?

The effect of permissions when removing files

Directory Write Permission



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

[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

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

Benji has read and write permission on myfile

Benji (and everyone else) has all permissions.

So why can't Benji remove his own file?

Answer:

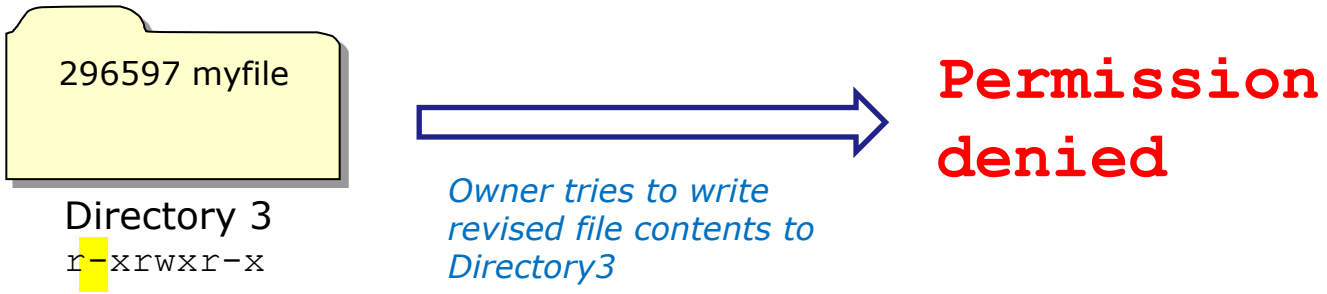
Removing a file requires write permission on the directory containing the file.

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



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

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

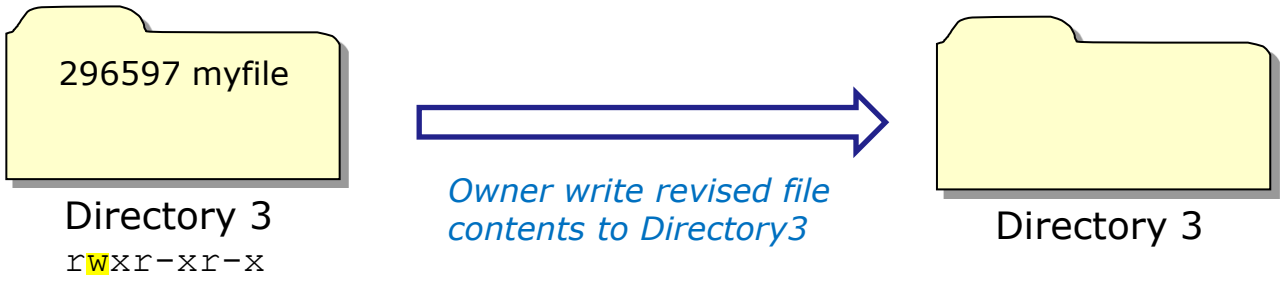
Now Benji has no permissions on this file

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
```

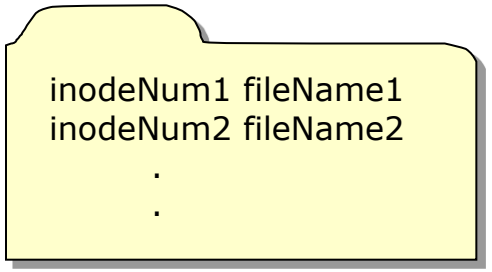
With write permission, Benji can remove any of the files from this directory … even the ones he does not have read & write permission for.



Directory Permissions

Directory Read Permission

rwx



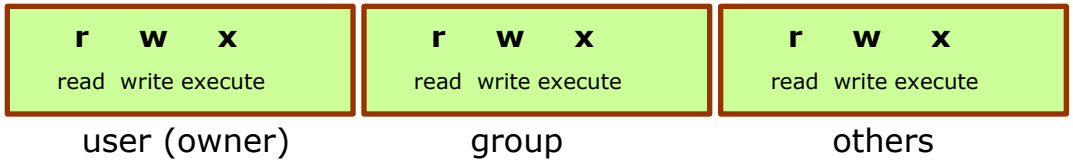
rwx

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

Removing directory READ permission

- can't list files in directory

Directory Read Permission

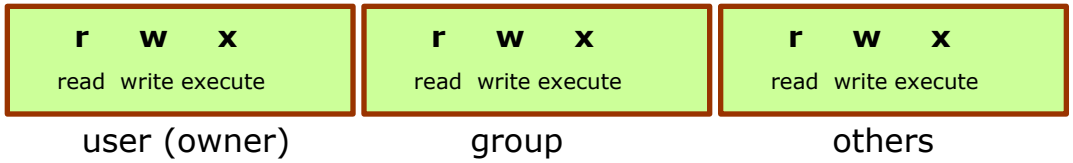


Start with normal directory permissions:

```
/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
```

If read permission is removed from the directory … can we still list the directory contents?

Directory Read Permission



Remove read permission and confirm it's gone

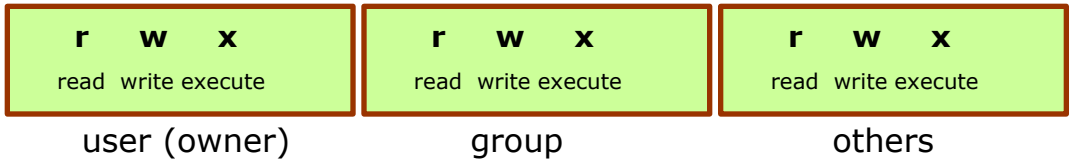
```
/home/cis90/roddyduk $ chmod u-r examples
/home/cis90/roddyduk $ ls -ld examples
d-wxrwxr-x 4 roddyduk cis90 4096 Oct 19 13:59 examples
```

Can we still list the directory contents?

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

NO!

Directory Read Permission

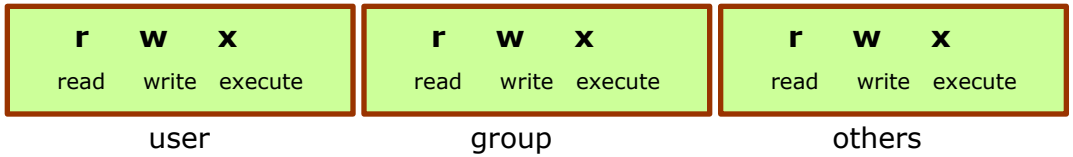


Start with normal directory permissions:

```
/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
```

*If read permission is removed from the directory … can we still **cd** into the directory?*

Directory Read Permission



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
```

*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 ...

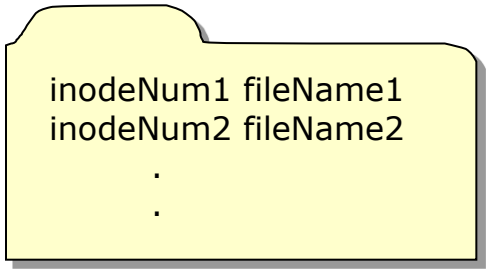
- *we still can't list the contents,*
- *yet we can still access anything in the directory!*

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.

The effect of WRITE permission on directories

Directory Write Permission

rwx



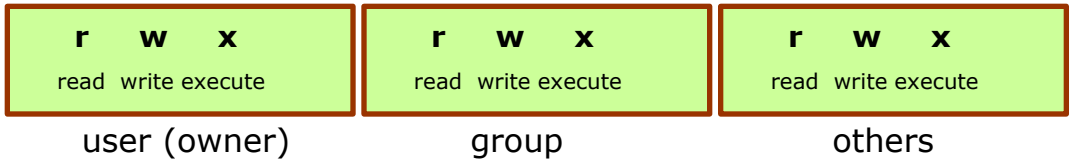
rwx

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 WRITE permission

- can't copy files to it
- can't remove files from it
- can't move files out of it
- can't add links to it

Directory Write Permission

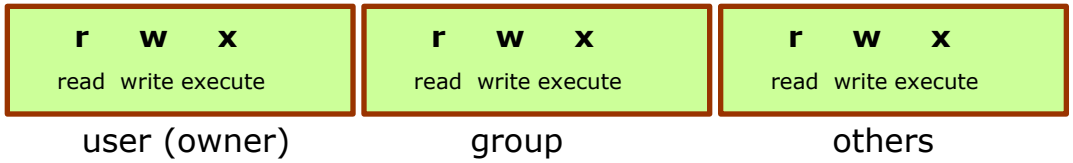


Start with normal directory permissions:

```
/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
```

*If write permission is removed from the directory … can we **remove files** from the directory?*

Directory Write Permission



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/
```

Can we remove files from the directory?

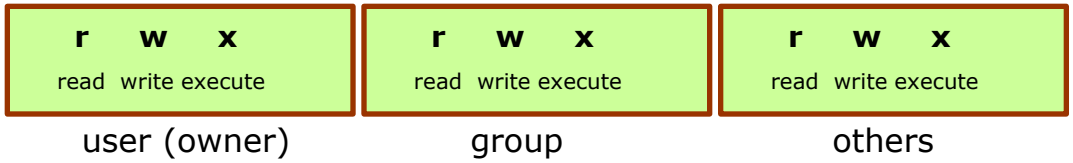
```
/home/cis90/roddyduk/examples $ rmdir dogs
rmdir: dogs: Permission denied
```

NO!

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

Yet we can still cd into and list directory contents

Directory Write Permission

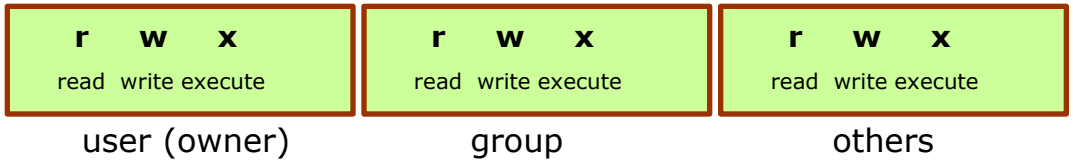


Start with normal directory permissions:

```
/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
```

*If write permission is removed from the directory … can we **create new files or copy/move files** into the directory?*

Directory Write Permission



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/
```

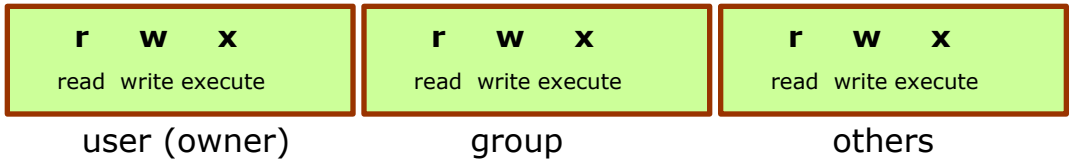
Can we create new files or copy/move files into the directory?

```
/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 Write Permission

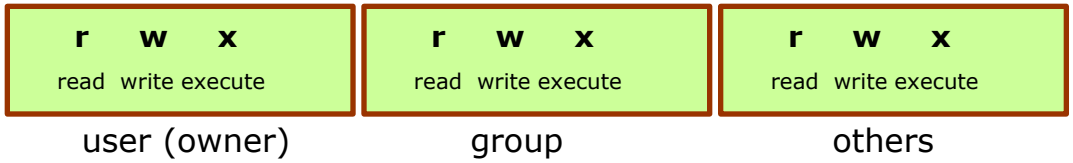


Start with normal directory permissions:

```
/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
```

If write permission is removed from the directory … can we move files out of the directory?

Directory Write Permission



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/
```

Can we move files out of the directory?

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

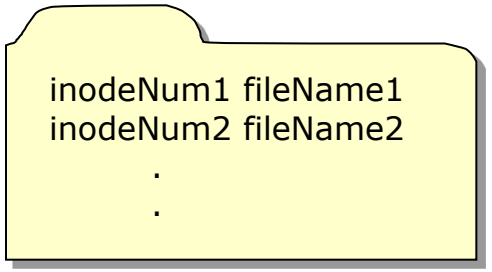
NO!

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

The effect of EXECUTE permission on directories

Directory Execute Permission

rwx



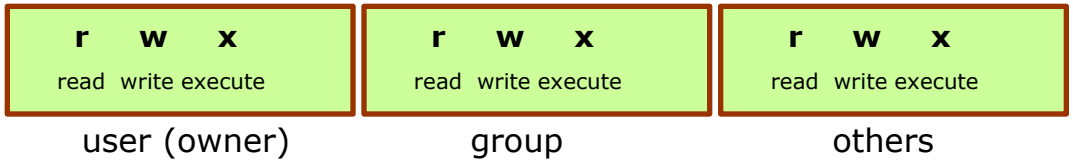
rwx

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 EXECUTE permission

- can't retrieve inode information (long listing) or data (content)
- can't cd into directory

Directory Execute Permission

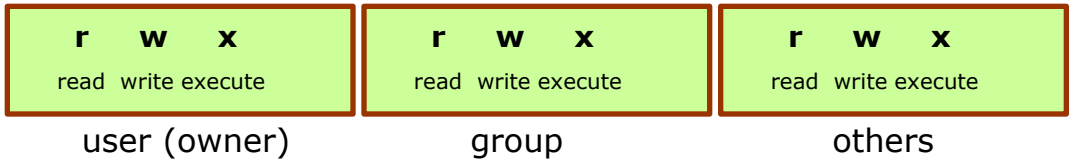


Start with normal directory permissions:

```
/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
```

If execute permission is removed from the directory … can we change into (cd) the directory?

Directory Execute Permission



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/
```

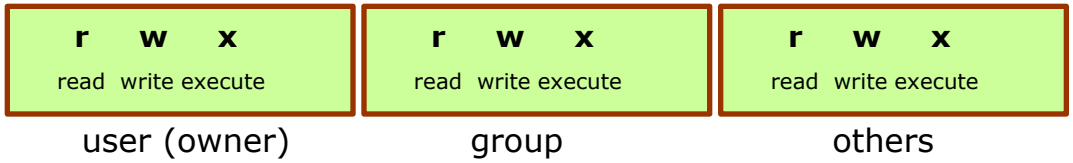
Can we change into (cd) the directory?

```
/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 Execute Permission

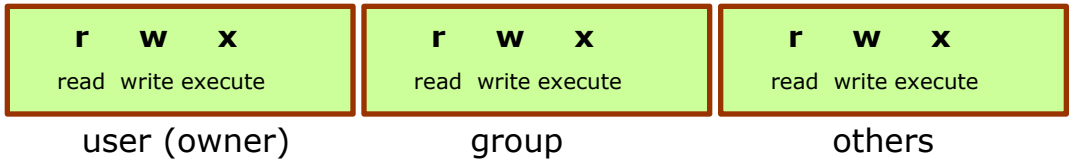


Start with normal directory permissions:

```
/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
```

If execute permission is removed from the directory … can we list directory contents?

Directory Execute Permission



Remove execute permission and confirm it's gone

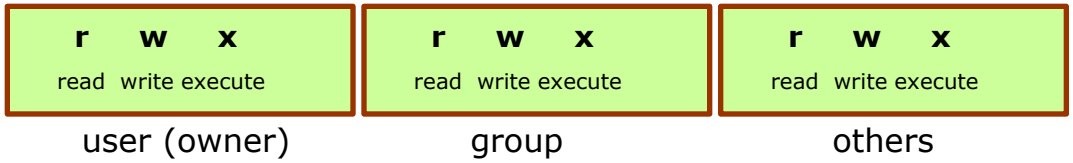
```
/home/cis90/roddyduk $ chmod u-x examples
/home/cis90/roddyduk $ ls -ld examples
drw-rwxr-x 4 roddyduk cis90 4096 Oct 19 13:59 examples/
```

Can list directory contents?

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

Yes

Directory Execute Permission

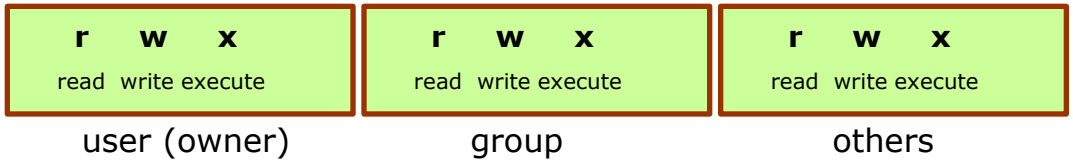


Start with normal directory permissions:

```
/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
```

If execute permission is removed from the directory … can we do a long listing of the directory?

Directory Execute Permission



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/
```

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

Used for setting the default permissions on new files and directories

umask – user file-creation mask

Syntax:

umask [*mask*]

a bitmask used to strip permission bits off newly created files and directories

Examples:

- **umask**
- **umask 002**
- **umask 777**

If the mask is not specified, the current umask setting is displayed

File Permissions

Default Permissions

Default system permissions

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

When new files or directories are created they start with the default permissions above, then the current setting of the umask is applied to strip away any unwanted permissions.

For example, if the umask setting is:

777 – then all permissions are stripped off the default
000 – then no permissions are stripped off the default
022 - strips off just the write permissions from group and other users from the default

File Permissions

umask - examples

With no argument, the current umask setting is shown

```
[simmsben@opus Directory3]$ umask
0002
```

this umask setting will strip write permission from Others

```
[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-	*default system permissions for a file*
002	-------w-	*umask setting (strips these permissions from default)*
664	rw-rw-r--	*result after masking*

File Permissions

umask - examples

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

Change umask to 000

```
666   rw-rw-rw-
000   ---------
666   rw-rw-rw-
```

default system permissions for a file

umask setting (strips these permissions from default)

result after masking

File Permissions

umask - examples

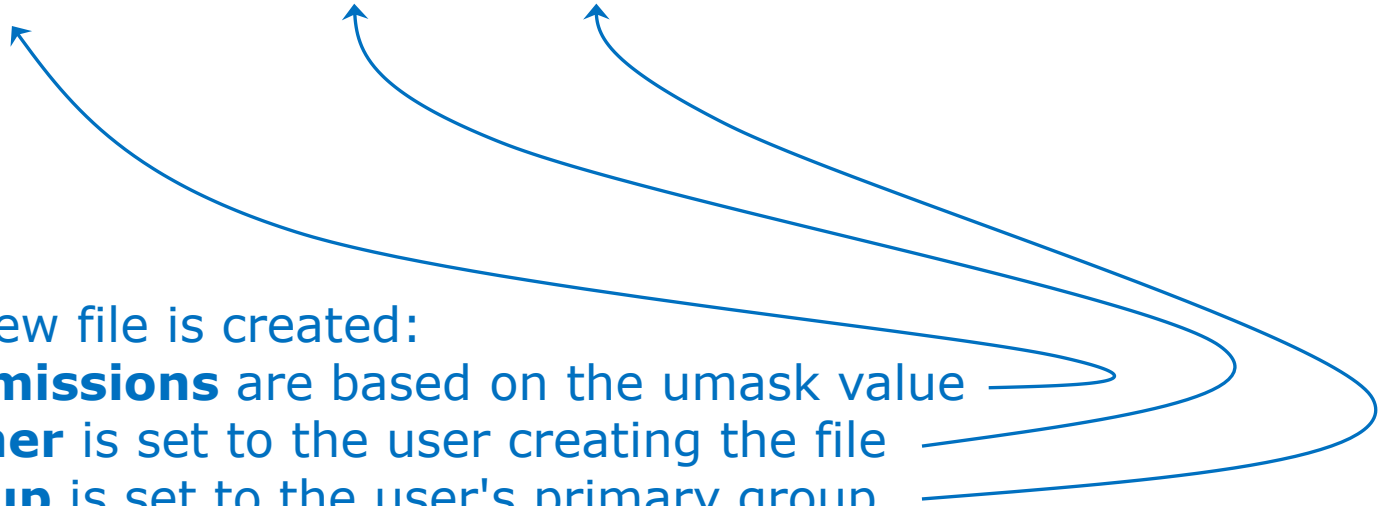
```
[simmsben@opus Directory3]$ umask 022
[simmsben@opus Directory3]$ rm myfile
[simmsben@opus Directory3]$ touch myfile
[simmsben@opus Directory3]$ ls -l
total 4
-rw-r--r-- 1 simmsben cis90 0 Oct 15 15:00 myfile
```

Change umask to 022

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

Cabrillo College

CIS 90 Linux Lab Exercise

Lab 6: File Permissions

Spring 2012

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	change file's group
chmod	change file permissions
chown	change file owner (superuser only)
groups	show group membership
stat	show all file inode information
umask	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.

Lab 6 Five Posts

Quiz questions for next class:

- 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