

Lesson Module Status

- Slides - draft
- Flash cards –
- properties – done
- 1st minute quiz – done
- Web Calendar summary –
- Web book pages -
- Commands –
- Lab tested –

- Backup headset charged -
- CCC Confer wall paper - done
- Slides & Lab uploaded -



Dennis



Sean



Christopher



Francisco



Rich

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



Salena



Abd



Sarah



Astitow



Mike D.



Alex



Christine



Steven



Richie



Nathan



Tony



James G.



Sergio



Anthony



Fernando



Miguel



Lars



Jennifer



Rudy



Laura P.



Nick



Juan



Jacob



Andrew



Luke

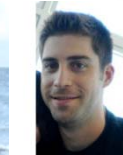


Saulius

Online Class Students



Edtson



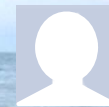
James B.



Liz



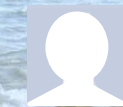
Casady



Jason



Dale



Aaron



Steve



Matt



Songul



Stephanie



Victor



Tanya



Mike P.



Adriana



Laura S.



Olivia



Janelle

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

Quiz

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

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

email answers to: risimms@cabrillo.edu

(If you are in the classroom you can write your answers on a scrap piece of paper and hand it in)



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

File Permissions

Objectives	Agenda
<ul style="list-style-type: none">• 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	<ul style="list-style-type: none">• Quiz• Review test results• Question on previous material• File permissions• Wrap up

* = hands on exercise for topic



Housekeeping

Test 1 – Most Missed Questions

Question# - count of students with incorrect answers

QX2	43	/etc/passwd fields
Q8	34	path used by shell
Q14	29	shell filename expansion
QX1	25	type command
Q2	24	/etc/shadow
Q5	21	shell filename expansion
QX3	17	relative pathname
Q6	13	file types
Q15	12	bc and mail commands
Q12	11	inode number
Q11	11	setting prompt
Q3	11	creating new message in mail
Q10	8	file owner
Q4	7	pwd command
Q7	6	/etc/password absolute path
Q13	6	navigating with cd and ls commands
Q9	6	wc command
Q1	4	; metacharacter

Correct answers are in /home/cis90/answers

Test 1 – Most Missed Questions

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

```
/home/cis90/simmsben $ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
< snipped >
ojedavic:x:1252:90:Victor Ojeda:/home/cis90/ojedavic:/bin/bash
millehom:x:1253:90:Homer Miller:/home/cis90/millehom:/bin/sh
alvesdes:x:1254:172:Desmond Alves:/home/cis172/alvesdes:/bin/bash
< snipped >
mikedel:x:1276:172:Delfino Mike:/home/cis172/mikedel:/bin/bash
seanbla:x:1277:172:Black Sean:/home/cis172/seanbla:/bin/bash
veracroc:x:1278:172:Roche Verachai:/home/cis172/veracroc:/bin/bash
/home/cis90/simmsben $
```

The answer



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

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

Test 1 – Most Missed Questions

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

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

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

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

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

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

Answer: no

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

Test 1 – Most Missed Questions

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

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

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

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

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

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

Answer

Test 1 – Most Missed Questions

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

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



The dig command is in the /usr/bin directory

Answer

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

```
[
```

```
4lltoppm
```

```
< snipped >
```

```
diffstat
```

```
dig
```

```
dir
```

```
< snipped >
```

```
nroff
```

```
nsgmls
```

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

```
nslookup
```

```
nspr-config
```

```
pnmalias
```

```
pnmarith
```

```
pnmcat
```

```
zsoelim
```

Test 1 – Most Missed Questions

[Q2] In what file are all passwords stored?

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

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

Answer: /etc/shadow

Note: this was a Flashcards question

Card 8 of 22 (front and back)

In what file are all the account passwords stored?

`/etc/shadow`

Reference

Test 1 – Most Missed Questions

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

Answer: echo B????e

Note: this was a Flashcards question

Card 2 of 4 (front and back)

What echo command would show in the working directory any 5 letter filenames starting with a B and ending with e?

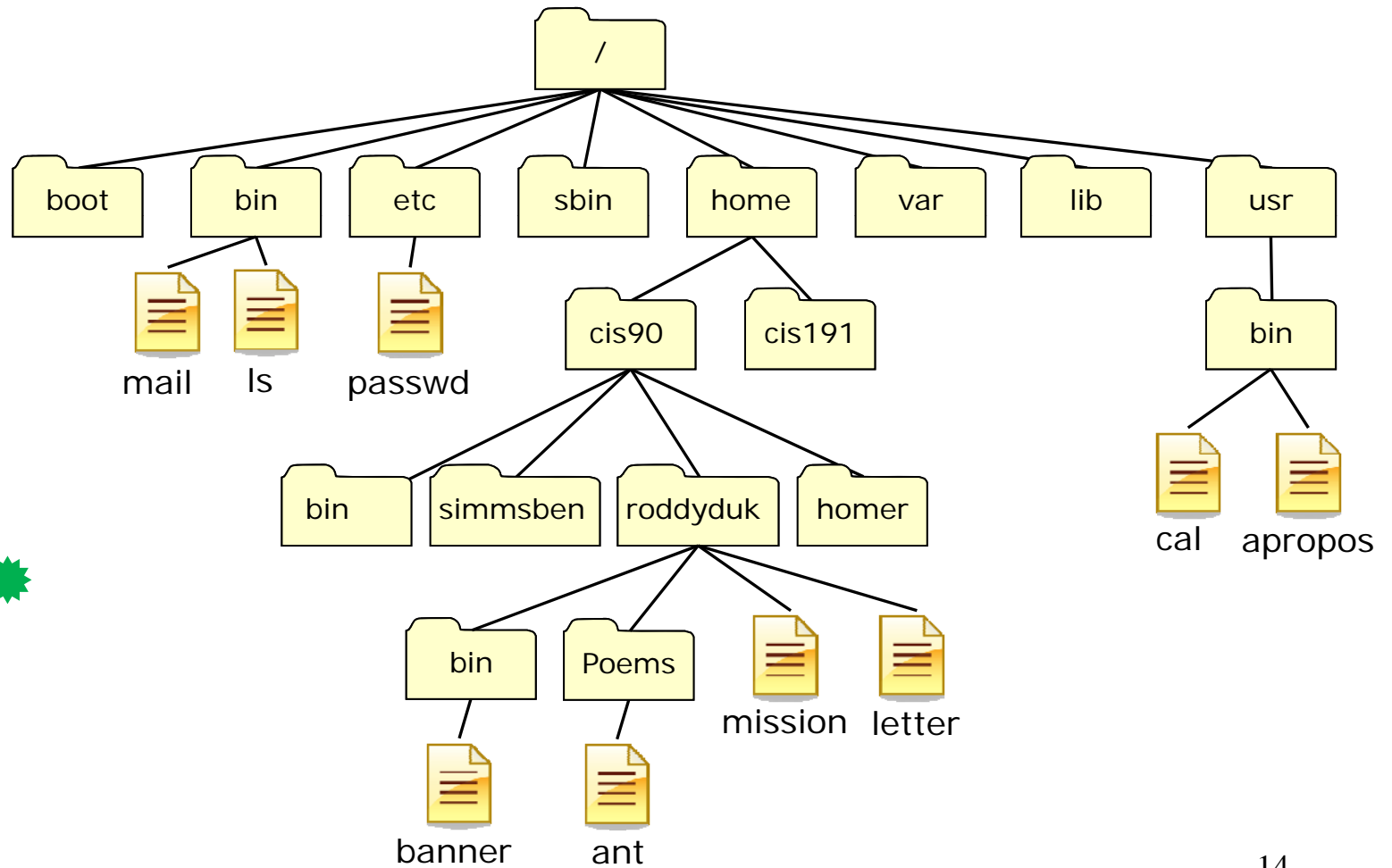
echo B????e

[Reference](#)

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

UNIX File Tree

Pathname Practice



Previous material and assignment

1. Questions on last Lab assignment?
2. Questions on file management commands?

one more
mv

Home directory cleanup

- Move your graded work to /class/labs

```
mv test01.graded class/exams/
```

- Check with ls command

Permissions

File Permissions

Relevant fields from the inode

How do we control access to files and directories?

Answer: file permissions

File Permissions

Relevant fields from the inode

What permissions are there?

Answer: read, write and execute

File Permissions

Relevant fields from the inode

Who do permissions apply to?

Answer:

*The **owner**,
a **group**
and everyone else (**others**)*

File Permissions

Relevant fields from the inode

```

/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

```

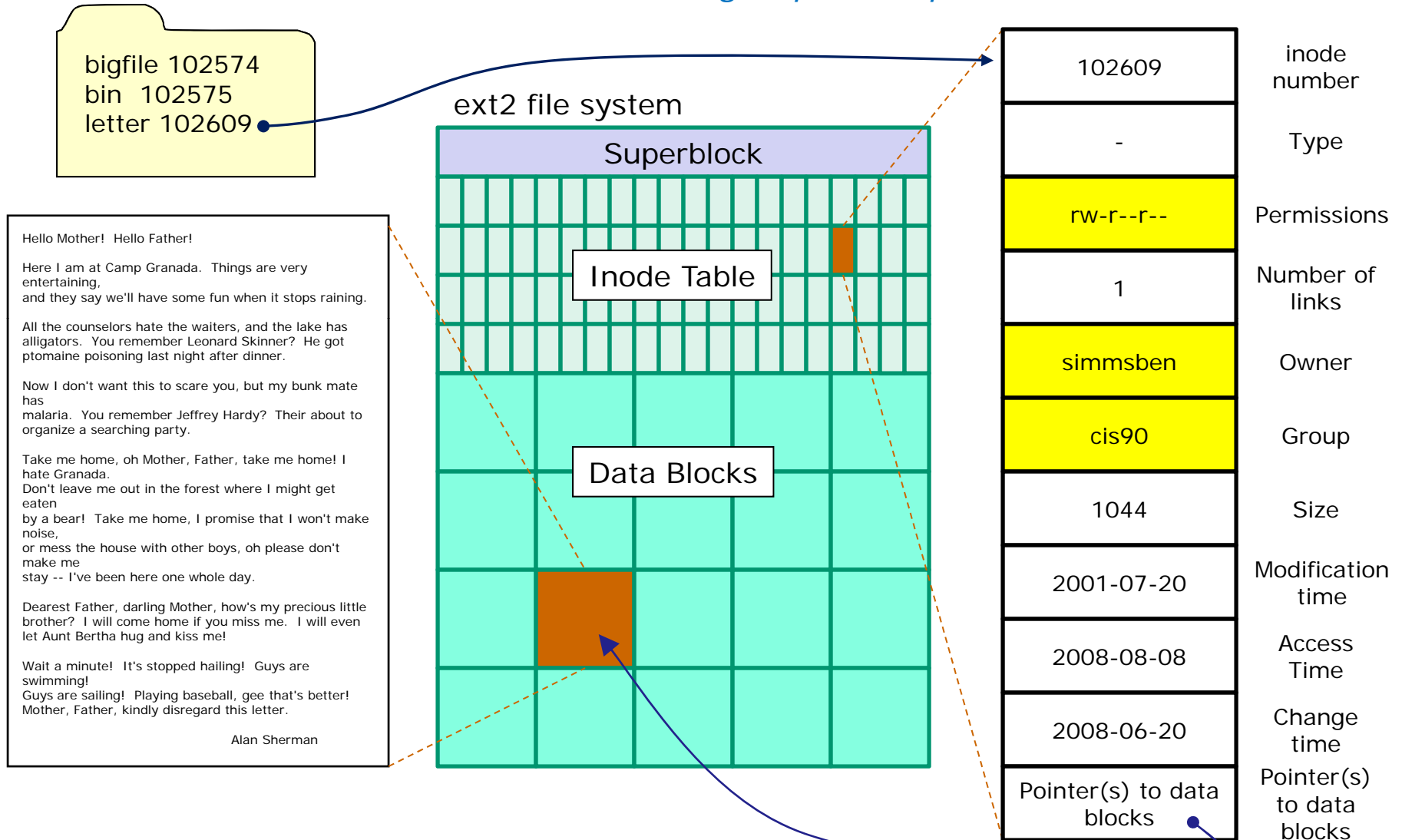
Permissions (points to the first three columns of the file listing)

Owner (points to the owner column)

Group (points to the group column)

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

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



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

File Permissions

Every user has a unique user ID (uid)

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


File Permissions

Every user belongs to a primary group (gid)

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

File Permissions

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

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

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

File Permissions

Relevant fields from the inode

Permissions are shown as triplets using either letters or numbers

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

rwx	rwx	rwx
owner	group	other

These permissions would be 777 using numeric form

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

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

File Permissions

Example: letter file

```

/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

```

Diagram annotations:

- Permissions* points to the first column of the output.
- Owner* points to the second column of the output.
- Group* points to the third column of the output.

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

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 ~]$
```

numeric form

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

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

File Permissions

Binary

Decimal	Binary
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Permissions are stored internally using binary numbers

File Permissions

Viewing relevant fields from the inode

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

110 | 100 | 100
 |
`rw-r--r--`

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

110 (binary) = 6 (decimal)
100 (binary) = 4 (decimal)

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

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 `rxwxrw-r--?`

File Permissions

7 6 4
111110100

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

Answer: 764

File Permissions

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

File Permissions

7 5 5
111101101

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

Answer: 755

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

File Permissions

What permissions are 644?

File Permissions

What permissions are 644?

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

Answer:

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

File Permissions

Relevant Commands

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

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

chmod - Changes the permissions of a file.

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

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

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

chown

File Permissions

chown and chgrp command - examples

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

*Only root can use **chown***

chgrp

File Permissions

chown and chgrp command - examples

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

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

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

*Everyone can use **chgrp***

chmod

File Permissions

chmod command - examples

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

```
/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 - examples

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

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

remove execute from all

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

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

*add execute to others
and group*

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

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

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

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

Use chmod to add or remove permissions from a file

File Permissions

chmod command - examples

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

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

File Permissions

chmod command - examples

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

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

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

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

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

File Permissions

Commands that require file permissions

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

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
```

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

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
```

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

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
```

What happens now when you type myfile?

rm <file>

and

permissions

File permissions and rm command example 1

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

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

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

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

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

*Permissions on myfile
are 744. Owner has
all permissions.*

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

So why can't Benji remove his own file?



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

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



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

File permissions and rm command

Example 2

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

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

Now Beni has no permissions on this file

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

So how come he can delete it?



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

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



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

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

bigfile 102574
bin 102575
letter 102609

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.

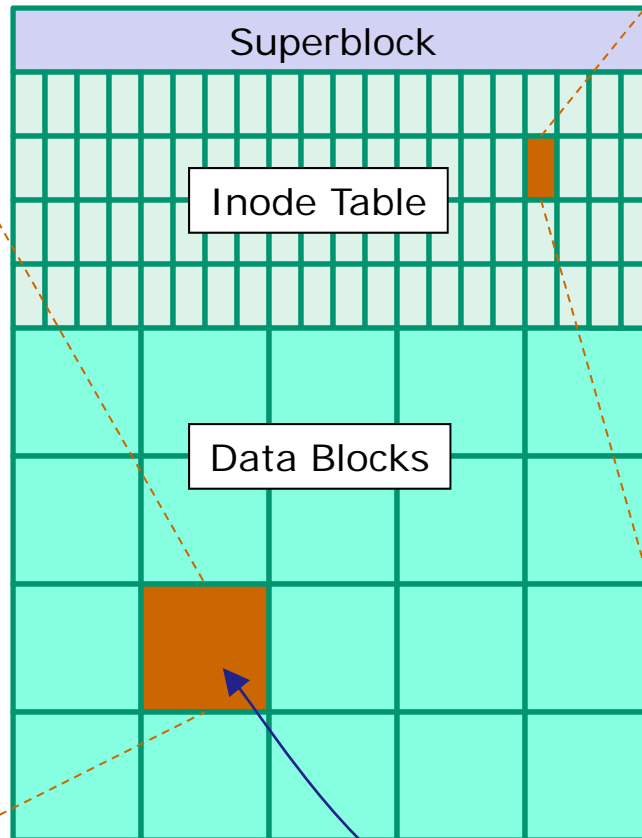
Take me home, oh Mother, Father, take me home! I hate Granada.
Don't leave me out in the forest where I might get eaten by a bear! Take me home, I promise that I won't make noise, or mess the house with other boys, oh please don't make me stay -- I've been here one whole day.

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

ext2 file system



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
```

umask

File Permissions

Default Permissions

What permissions are files created with?

Answer: This is determined by the umask setting

Default permissions

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

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

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

File Permissions

umask - examples

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

```
666
-002
664
rw_rw_r__
```

```
666
-000
666
rw_rw_rw_
```

```
666
-022
644
rw_r__r__
```

Examples setup

File Permissions

exercise - setup



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

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

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

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

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


File Permissions

exercise - setup



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

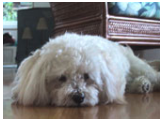
Note that other users have no permissions for the /home/cis90 directory

Examples owner write to files

File Permissions

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

Can Benji write to file1?



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

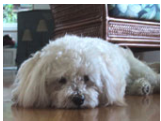
/home/cis90/simmsben/Directory1 $ echo "changes" > file1
```




File Permissions

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

Yes, he can. Can Benji write to file2?



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

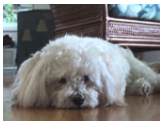
```
/home/cis90/simmsben/Directory1 $ echo "changes" > file1 
/home/cis90/simmsben/Directory1 $ echo "changes" > file2
```





File Permissions

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

Yes, he can. Can Benji write to file3?



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

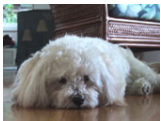
```
/home/cis90/simmsben/Directory1 $ echo "changes" > file1 
/home/cis90/simmsben/Directory1 $ echo "changes" > file2 
/home/cis90/simmsben/Directory1 $ echo "changes" > file3
```



File Permissions

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

No he cannot



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

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



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

Benji could write to file1 and file2 but not file3

Examples group write to files

File Permissions

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

Can Duke write to Benji's file1?



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



File Permissions

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

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



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



File Permissions

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

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



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



File Permissions

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

No, he cannot.



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



Examples other write to files

File Permissions



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

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



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

File Permissions



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

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



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

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

Examples
owner
remove files

File Permissions

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



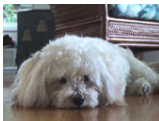
```
[simmsben@opus Directory1]$ ls -ld
drwxr-x--x 2 simmsben cis90 4096 Oct 15 12:42 .
[simmsben@opus Directory1]$
[simmsben@opus Directory1]$ ls -l
total 24
-rw-rw-r-- 1 simmsben cis90 15 Oct 15 11:13 file1
-rwxr-xr-x 1 simmsben cis90 15 Oct 15 11:13 file2
-r-xr-xr-- 1 simmsben cis90 15 Oct 15 11:13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

Can Benji remove his own files?

File Permissions

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

Can Benji remove his files?



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



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

Examples

group

remove files

File Permissions

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



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

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



Can Duke remove Benji's files?

File Permissions

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



Can Duke remove Benji's files?

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

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



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

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





Examples
other
remove files

File Permissions

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



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

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



Can Mary remove Benji's files?

File Permissions

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



Can Mary remove Benji's files?

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

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



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

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

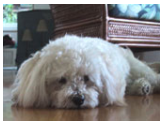
Examples

owner

read files

File Permissions

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



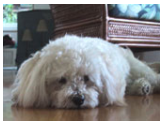
```
[simmsben@opus Directory1]$ ls -ld
drwxr-x--x 2 simmsben cis90 4096 Oct 15 12:42 .
[simmsben@opus Directory1]$
[simmsben@opus Directory1]$ ls -l
total 24
-rw-rw-r-- 1 simmsben cis90 15 Oct 15 11:13 file1
-rwxr-xr-x 1 simmsben cis90 15 Oct 15 11:13 file2
-r-xr-xr-- 1 simmsben cis90 15 Oct 15 11:13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

Can Benj read his own files?

File Permissions

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

Can Benj read his own files?



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

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



yes he can

Examples

group

read files

File Permissions

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



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



Can Duke read Benji's files?

File Permissions

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



Can Duke read Benji's files?

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



Yep!

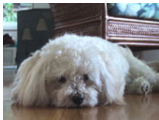
```
[roddyduk@opus ~]$ cat ../simmsben/Directory1/file*
blah blah blah
blah blah blah
blah blah blah
[roddyduk@opus ~]$
```



Examples other read files

File Permissions

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



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

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

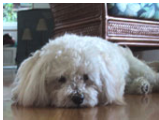


Can Mary read Benji's files?

File Permissions

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

Can Mary read Benji's files?



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

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



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

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



Examples

owner

execute files

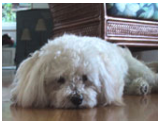
File Permissions

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

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

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

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



Can Benji run his own files?

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

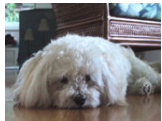
File Permissions

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

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

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

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



Can Benji run his own files?

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

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



He can for ones he has execute permission for

Examples

group

execute files

File Permissions

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



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



Can Duke run Benji's files?

File Permissions

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



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



Can Duke run Benji's files?

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



He can for the ones he has execute permission for

Examples other execute files

File Permissions

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



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

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



Can Mary execute Benji's files?

File Permissions

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



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

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



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

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



Lab 6

The screenshot shows a web browser window with the following content:

- Browser Tabs:** Google, (0 unre...), Rich's..., Cabrill..., Frontie..., CIS 90 ...
- Address Bar:** file:///H:/data-simms-teach.com/docs/cis90/cis90lab06.html
- Page Header:** Cabrillo College logo
- Main Content:**
 - CIS 90 Linux Lab Exercise**
 - Lab06: File Permissions**
 - Fall 2010**
- Objectives:**

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

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

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

 - From your home directory, type the `ls -l` command.
Who owns these files? To which group do they belong?
How can you distinguish file entries from directory entries?
 - 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?
 - Now that you have copied the file `uhistory` to your home directory, who owns it? What are the permissions?
 - Display the contents of the file `uhistory` on your screen.
Now take away read permission using the command:

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

Wrap up

New commands:

chgrp

chmod

chown

umask

change file's group

change file permissions

change file owner (superuser only)

change permission mask

New Files and Directories:

NA

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