Revision lecture on UNIX - 2

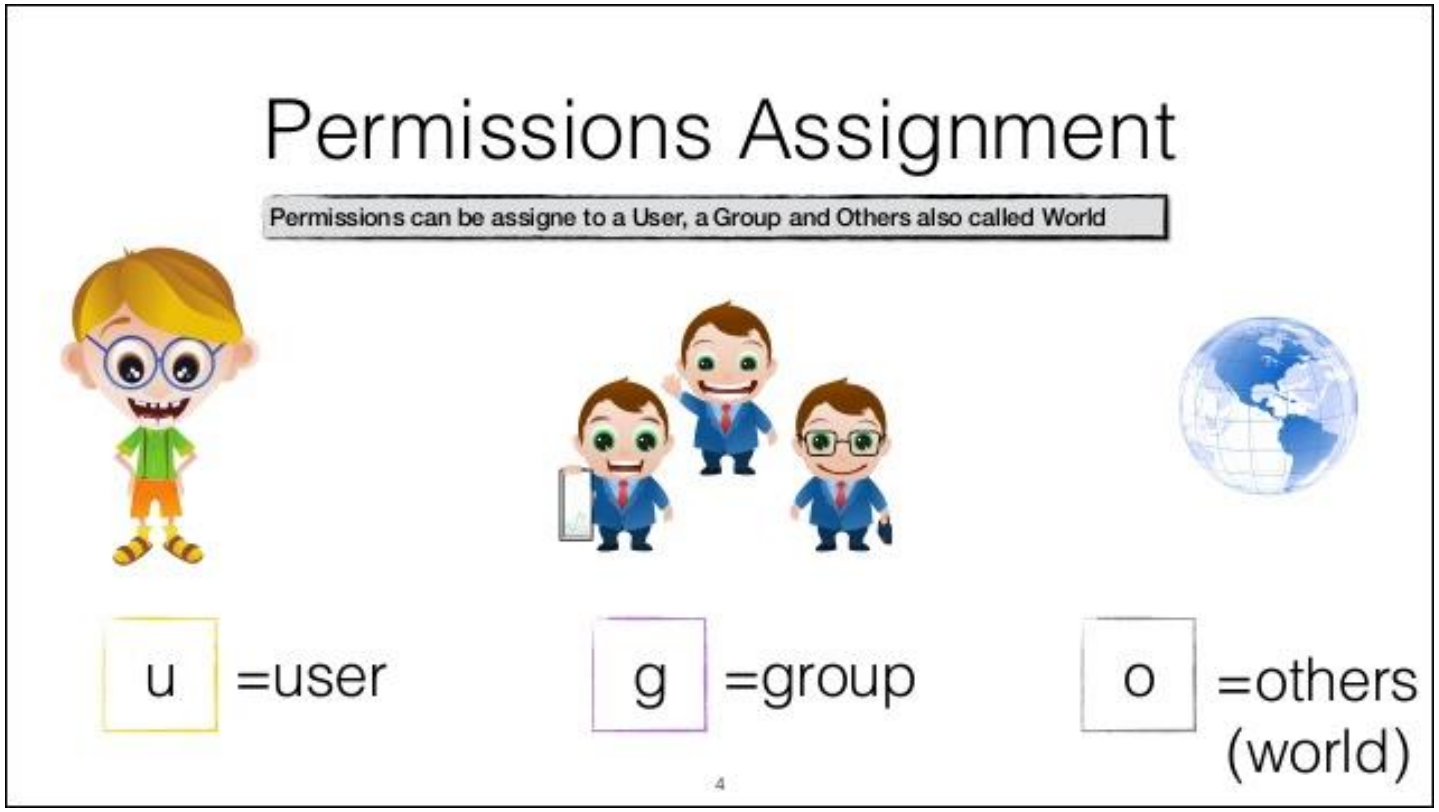
COMP1204: Data Management

Permissions on files

See slides at
https://secure.ecs.soton.ac.uk/notes/comp1204/2020/dst/UNIX4.pdf

Permissions on files

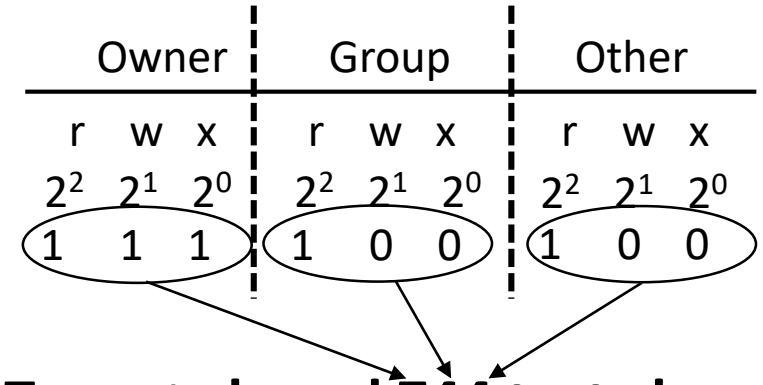
- The Unix file system provides permissions to restrict access to your files and directories.



https://www.slideshare.net/matteosetti/unix-userspermissionsslideshare

Changing permissions on files

- **chmod** takes a special argument of three parts (see table) to assign the desired rights to files or directories.



- Try out **chmod 744 test.sh**

File Type (character 1)	Owner Access (characters 2-4)	Group Access (characters 5-7)	Other Access (character 8-10)
- = regular file d = directory	r = read w = write x = execute	r = read w = write x = execute	r = read w = write x = execute

http://people.ischool.berkeley.edu/~kevin/unix-tutorial/section4.html

Bash scripts, grep, sed and awk

See slides at
https://secure.ecs.soton.ac.uk/notes/comp1204/2019/dst/UNIX5.pdf

Loops in Bash shell script

- Often, when writing your UNIX scripts, you would have to loop through
  - all the files in a directory, or
  - through all the lines in a text file.

Construct to loop though all files in a directory

T

- *$var* is syntax to pass a variable value to a command.
- A variable lacks this $ sign if it is being assigned rather than referenced.

wildcard indicates contents of directory

- for *var* in *directory*/*; do <something with $*var*>; done

Start with **for**

Specify what is to be done in the loop

Terminate **for** loop

Example:

dst1m17@login:/home/dst1m17> **for i in /home/dst1m17/*; do echo $i; done**

Construct to loop though all the lines in a text file

- What about looping through all the lines in a text file?

produce the contents for the while loop

- cat file-name | while read line; do <something with $line>; done

Start with **while**

Specify what is to be done in the loop

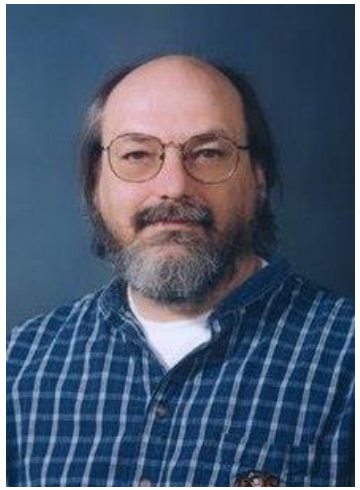
Terminate **while** loop

Example:

dst1m17@login:/home/dst1m17> **cat tomslee* | while read p; do echo $p; done**

grep, sed, awk

grep


Ken Thompson

- grep is a command to search input given to it.
- It looks for lines in the input that match a particular pattern or regular expression
- Usage: **grep** ***pattern input***
- The pattern can be constructed from regular expressions.

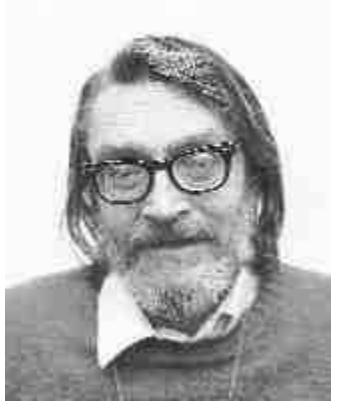
Special Characters in Regular Expressions & their meanings

Character	Meaning	Example
*	Match **zero, one or more** of the previous	`Ah*` matches "Ahhhhh" or "A"
?	Match **zero or one** of the previous	`Ah?` matches "Al" or "Ah"
+	Match **one or more** of the previous	`Ah+` matches "Ah" or "Ahhh" but not "A"
\	Used to **escape** a special character	`Hungry\?` matches "Hungry?"
.	Wildcard character, matches **any** character	`do.*` matches "dog", "door", "dot", etc.
( )	**Group** characters	See example for `|`
[ ]	Matches a **range** of characters	`[cbf]ar` matches "car", "bar", or "far" `[0-9]+` matches any positive integer `[a-zA-Z]` matches ascii letters a-z (uppercase and lower case) `[^0-9]` matches any character not 0-9.
|	Matche previous **OR** next character/group	`(Mon)|(Tues)day` matches "Monday" or "Tuesday"
{ }	Matches a specified **number of occurrences** of the previous	`[0-9]{3}` matches "315" but not "31" `[0-9]{2,4}` matches "12", "123", and "1234" `[0-9]{2,}` matches "1234567..."
^	**Beginning** of a string. Or within a character range `[]` negation.	`^http` matches strings that begin with http, such as a url. `[^0-9]` matches any character not 0-9.
$	**End** of a string.	`ing$` matches "exciting" but not "ingenious"

https://github.com/aloverso/SoftwareSystems/wiki/Deep-Dive-Haley-Grep,-Sed,-Awk

sed

- sed is a text stream editor
- Reads input and modifies it as specified by a list of commands. The modified input is then written to the standard output.
- Usage: **sed [options] command [file ...]**
- Most commonly used command is to substitute text with something else



Lee E. McMahon

awk

- A utility for processing structured text files.
- Sees the text file as rows and columns of data in a table
- Good to parse the tables and do some calculations on the parsed output

Brian Kernighan

Example MCQs for UNIX part of COMP1204 exam

Example question 1

You have compiled your C code to produce a binary file named *print-attendance.o*. Set the permissions for the following:

> You have full-access to the file.
>
> The group can only read and execute the file.
>
> Everyone else is allowed to only execute it.

1. chmod 751 *print-attendance.o*
2. chmod 640 *print-attendance.o*
3. chmod 740 *print-attendance.o*
4. None of the above

Example question 1 - answer

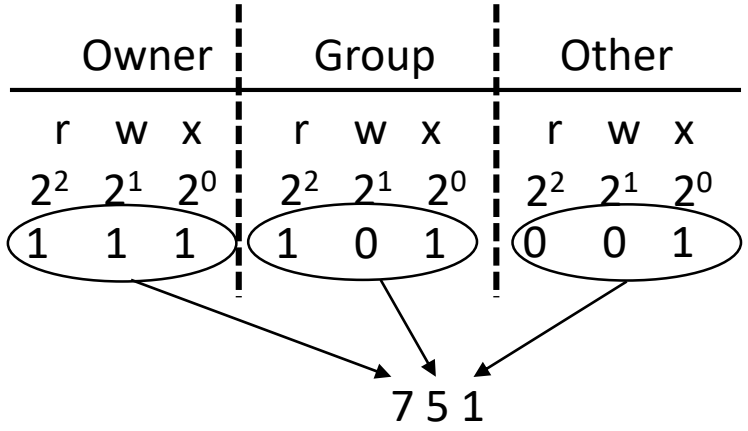
You have compiled your C code to produce a binary file named *print-attendance.o*. Set the permissions for the following:

You have full-access to the file.

The group can only read and execute the file.

Everyone else is allowed to only execute it.

1. **chmod 751 *print-attendance.o***
2. chmod 640 *print-attendance.o*
3. chmod 740 *print-attendance.o*
4. None of the above



Example question 2

In your empty home directory you run the command

touch record-a.txt record-A.txt record-A.TXT RECORD-A.txt reCORD-A.txt

Now what will be the output of the following:

file * | wc –l

1. 1 as all the files have the same name because UNIX filenames are not case-sensitive
2. 5
3. file command not found

Note: The -l switch to wc prints the number of lines

Example question 2 - answer

Q. In your empty home directory you run the command

touch record-a.txt record-A.txt record-A.TXT RECORD-A.txt reCORD-A.txt

Now what will be the output of the following:

file * | wc –l

1. 1 as all the files have the same name because UNIX filenames are not case-sensitive
2. **5**
3. file command not found

Example question 3

You have a small text file named *samples.txt* with 79 lines of text in it.

How would you create another file *output.txt* containing 15 repetitions of the last 15 lines of *samples.txt*

1. for i in {1..15}; do tail -n 15 *samples.txt*; done > *output.txt*
2. for i in {1..15}; do cat *samples.txt*; done > *output.txt*
3. for i in {1..79}; do tail -n 15 *samples.txt*; done < *output.txt*

Example question 3 - answer

You have a small text file named *samples.txt* with 79 lines of text in it.

How would you create another file *output.txt* containing 15 repetitions of the last 15 lines of *samples.txt*

1. **for i in {1..15}; do tail -n 15 *samples.txt*; done > *output.txt***
2. for i in {1..15}; do cat *samples.txt*; done > *output.txt*
3. for i in {1..79}; do tail -n 15 *samples.txt*; done < *output.txt*

- Please email me if you have any questions – dst1m17@soton.ac.uk
- Other revision sessions: Recordings will be posted next week. Keep an eye on the noteswiki page.
- Converting binary to decimal https://www.rapidtables.com/convert/number/how-binary-to-decimal.html