Tutorial4: Data Representation / Numbering Conversion / File Permissions

CDOT
SICT AR Meeting Area
People

get involved with CDOT

as a Student
as an Open Source Community Member
as a Company

courses

BTC640
BTH740
BTP300
DPI908
DPS901
DPS905
DPS909
DPS911
DPS914
DPS915
DPS924
DPS931
EAC234
ECL500
GAM531
GAM666
GAM670
GPU610
LUX Program
MAP524
OOP344
OPS235
OPS245
OPS335
OPS345
OPS435
OPS445
OPS535
OPS635
OSD600
OSD700
OSL640
OSL740
OSL840
Real World Mozilla
RHT524
SBR600

Data Representation / Numbering Conversion / File Permissions

Main Objectives of this Practice Tutorial

- Understand how digital computers store data (i.e. data representation)
- Define **decimal**, **binary**, **octal** and **hexadecimal** numbers
- Manually perform **numbering conversions** between the **decimal**, **binary**, **octal** and **hexadecimal** numbering systems
  (without the use of a computer or calculator)
- Explain the purpose of **file permissions**
- Explain how permissions work differently for **directories** as opposed for **regular files**
- Change file **permissions** with the **chmod** command (both *symbolic* and *absolute* methods)
- Use the **umask** command to automatically assign permissions for **newly created directories** and **regular files**

Tutorial Reference Material

Course Notes	Numbering Conversion / File Permissions Reference	YouTube Videos

SEC520
SPO600
SRT210
ULI101

course projects

Course Project List
Potential Course Projects
Contrib Opportunities

links

CDOT
Planet CDOT
FSOSS

Tools

What links here
Related changes
Special pages
Printable version
Permanent link
Page information

Slides:

- Week 4 Lecture 1 Notes: PDF | PPTX
- Week 4 Lecture 2 Notes: PDF | PPTX

Data Representation Definitions:

- Data Representation
- Decimal Numbers
- Binary Numbers
- Octal Numbers
- Hexadecimal Numbers

File Permission Concepts:

- Introduction to File Permissions

File Permission Commands:

- chmod
- umask

Instructional Videos:

- Numbering Conversions
- File Permissions

KEY CONCEPTS

Data Representation

Digital computers are **electronic devices** that contain a series of **circuits** and voltage levels that can store / represent data.

Binary numbers can represent those series of circuits with voltage levels.
Those binary numbers are combined in a sequence to form a **byte**. Bytes are used to represent numbers or characters.

DEC.	BINARY								HEX.
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1	1
2	0	0	0	0	0	0	1	0	2
3	0	0	0	0	0	0	1	1	3
4	0	0	0	0	0	1	0	0	4
5	0	0	0	0	0	1	0	1	5
6	0	0	0	0	0	1	1	0	6
7	0	0	0	0	0	1	1	1	7
8	0	0	0	0	1	0	0	0	8
9	0	0	0	0	1	0	0	1	9
10	0	0	0	0	1	0	1	0	A
11	0	0	0	0	1	0	1	1	B
12	0	0	0	0	1	1	0	0	C
13	0	0	0	0	1	1	0	1	D
14	0	0	0	0	1	1	1	0	E
15	0	0	0	0	1	1	1	1	F
16	0	0	0	1	0	0	0	0	10
17	0	0	0	1	0	0	0	1	11
.....									
.....									
.....									
253	1	1	1	1	1	1	0	1	FD
254	1	1	1	1	1	1	1	0	FE
255	1	1	1	1	1	1	1	1	FF

IT professionals may need to perform **numbering conversion** to use with
programming functions or *OS commands* to perform common operations on a computer system.

IT Professionals that Use Data Representation:

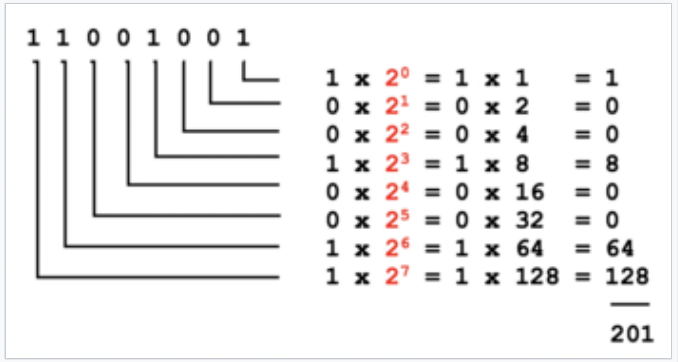
- *Network Specialists*: Building Large Networks via Sub-netting
- *Programmers:* Sending information over networks, files
- *Web Developers:* Setting color codes for webpage background or text
- *Unix/Linux System Administrators:* Setting *permissions* for files and directories

Numbering Conversion Methods

Method 1: Binary to Decimal

When converting **binary** numbers to **decimal** numbers, perform the following steps:

1. Write down the binary number.
2. Starting from the **right-side**, draw **L**'s below the binary number moving to the left (refer to diagram on right).



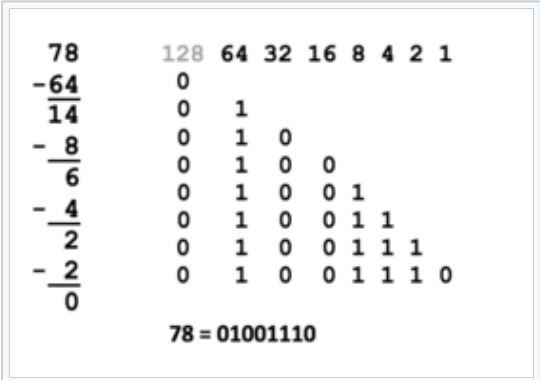
Performing a **binary** to **decimal** conversion.

3. Starting on the *rightmost* "**L**", multiply the value (placeholder) by **2** to the power of zero.
4. Continually repeat **step #3** moving leftwards, increasing the power of *2* by **1** (refer to diagram on right).
5. Add up the results to obtain the decimal value equivalent.

NOTE: To convert *octal* and *hexadecimal* numbers to **decimal**, replace the number **2** (in red in the diagram to the right) with **8** (for *octal*) or **16** (for *hexadecimal*).

Method 2: Decimal to Binary

When converting **decimal** numbers to **binary** numbers, perform the following steps:



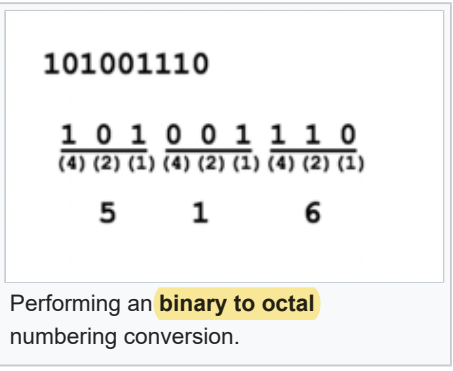
Performing a **decimal** to **binary** conversion.

1. Write down the **decimal number** to be converted.
2. On the *right-side*, write the number **1** and moving **leftwards**, keep doubling the numbers until that number is **greater than** the decimal number to be converted (refer to the diagram on the right).
3. Starting on the left-side of those doubled numbers, compare that number with the decimal number. If that number if less than or equal to the decimal number, then write a **1** below and subtract that number from the decimal number to get a remainder. If the number is greater than decimal number (or remainder), then write a **0** below.
4. Repeat **step #3** (moving rightwards and comparing the number with the decimal's remainder)

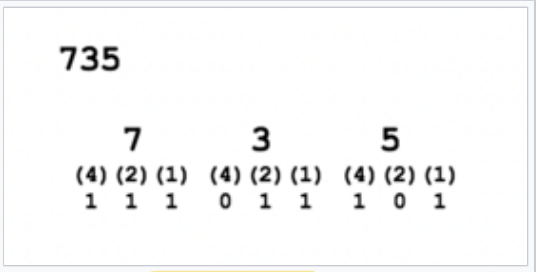
   NOTE: If you are converting to **8-bit**, **32-bit**, etc., add **leading zeros** if necessary.

Method 3: Octal to Binary / Binary to Octal

Binary to Octal



Performing an **binary to octal** numbering conversion.



Performing an **octal to binary** numbering conversion.

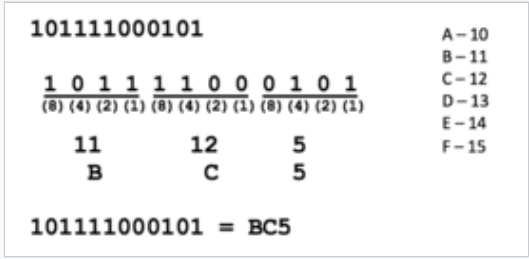
1. **One octal number** represents **3 binary numbers**, so **starting from right-side**, group binary digits into **groups of 3** (add leading zeros if necessary).
2. Write **(4)(2)(1)** under each **group of 3 binary numbers**.

3. Multiply the value or "placeholder" (i.e. **0**'s and **1**'s) by the corresponding **(4)(2)(1)** for each group to obtain the octal number (refer to diagram of *binary to octal* conversion).

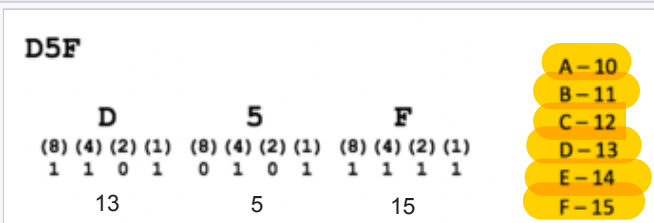
Octal to Binary

1. **One octal number** represents **3 binary numbers**, so space-out the octal numbers to make space for a binary number.
2. Write **(4)(2)(1)** under each octal number.
3. Write **0**'s or **1**'s for each group of binary numbers to add up to the corresponding octal number (refer to diagram of *octal to binary* conversion).

Method 4: Hexadecimal to Binary / Binary to Hexadecimal



Performing a **binary to hexadecimal** conversion.



Performing a **hexadecimal to binary** conversion.

Binary to Hexadecimal

1. **One hexadecimal number** represents **4 binary numbers**, so starting from right-side, group binary digits into **groups of 4** (add leading zeros if necessary).
2. Write **(8)(4)(2)(1)** under each group of 4 binary numbers.
3. Multiply the values or "placeholders" (i.e. **0**'s and **1**'s) by the corresponding (8)(4)(2)(1) for each group to obtain the octal number.
4. Convert values from **10** to **15** to **A** to **F** (refer to diagram of *binary to hexadecimal* conversion)

Hexadecimal to Binary

1. **One hexadecimal number** represents **4 binary numbers**, so space-out the hexadecimal numbers to make space for a binary number.
2. Convert letters **A** to **F** to **10** to 15 (refer to diagram of *binary to hexadecimal* conversion)
3. Write **(8)(4)(2)(1)** under each hexadecimal number.
4. Write **0**'s or **1**'s for each group of binary numbers to add up to the corresponding hexadecimal number (refer to diagram of *hexadecimal to binary* conversion).

Method 5: Octal to Hexadecimal / Hexadecimal to Octal

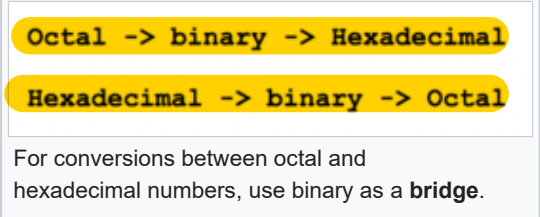
To convert using the method, simply use binary as a "**bridge**".

Example:

To convert octal to hexadecimal, convert octal to binary, then convert binary to hexadecimal.

To convert hexadecimal to octal, convert hexadecimal to binary, then convert binary to octal.

binary 作中間橋



For conversions between octal and hexadecimal numbers, use binary as a **bridge**.

File Permissions

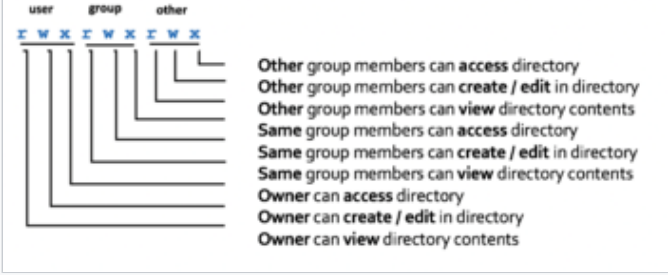
```
drwxr-xr-x 2 murray.saul users 6 Jan 19 14:06 mydir
-rw-r--r-- 1 murray.saul users 0 Jan 19 14:05 myregfile
```

Detailed directory listing showing permissions for a **directory** and a **regular file**.

Since Unix / Linux operating file systems allow for **multiple user accounts** it is essential to have a system to **share** or **limit** access to directories and files contained within the file system.

When **directories** and **regular files** are created, they are assigned to an **owner** (typically the username which is the creator). To *allow* or *limit* **access** to those files and directories, those files and directories are assigned to an initial **group** referred to as a "**primary group**".

Users that own those *directories* and *regular files* are referred to as **users**, users that belong within the same group are referred to as **same group members**, and those users that do NOT belong to a particular group are referred to as **other group members**.

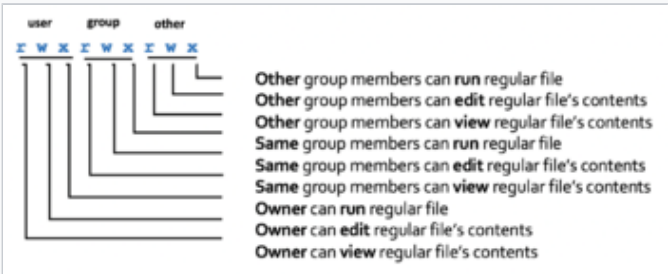


Permissions of a **directory** that contain subdirectories and regular files.

NOTE: In this course, we CANNOT create groups or assign users to groups in the **Matrix** server. Instead, you may learn how to those tasks when or if you take a Unix/Linux administration course. On the other hand, you can change which **user**, **same group members** or **other group members** can access or NOT access a directory or regular file.

File Permissions consist of **two-layers**:

First, the permissions of a **directory** that contains regular files, and **second**, permissions of the *subdirectories and/or regular files* within that directory.



Permissions of a **regular file** contained within a directory.

Permissions for directories have a different meaning than permissions for **regular files**. Refer to the diagrams to the right to see the explanation of permissions and how they differ between a directory and a regular file.

A symbol *dash* "**-**" indicates that the permission has **NOT** been granted.

The permissions of **newly-created** directories and regular files are automatically assigned via a **user mask** (we will discuss this shortly). In order to change permissions for directories and regular files, you would use the **chmod** command.

Changing File Permissions with "chmod" command:

Command	Description
`chmod ugo+x script.bash`	Add execute permissions to the file **script.bash** so it can be run.
`chmod u=rwx,go=x ~`	Set "**pass-thru**" permissions of your **home** directory for same group members and other group members to navigate to other subdirectories (that may have access / view permissions).
`chmod go-w ~/shared`	Remove write permissions for same group members and other group members for the directory **~/shared**
`chmod a=rx myfile.txt`	Set read and execute permissions for the directory **myfile.txt**

Examples of adding, removing and setting permissions using the **chmod** command with the **Symbolic** method.

Symbolic Method:

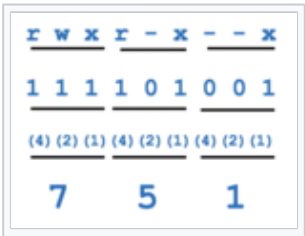
The chmod can use **symbols** to *add*, *remove*, and *set* **`rwx`** permissions for the **user**, **same group members**, and/or o**ther group members** for a directory or reqular file.

Octal (Absolute) Method:

You can also use **octal numbers** to **set** permissions. This method is a short-cut and may require less typing than using the *symbolic* method. You can only use this method to set file permissions (as opposed to *add* or *remove* permissions.

Since 1 octal digit represents 3 binary digits, one octal digit can represent the **`rwx`** permission granted or NOT granted. The permissions **rwx** are be in the form of 3 binary digits (1 represents the permission granted and 0 represents the permission NOT granted).

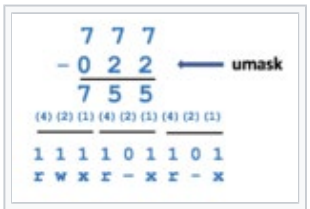
NOTE: You can use the **-R** option to set permissions for directory, subdirectory and directory contents **recursively**.



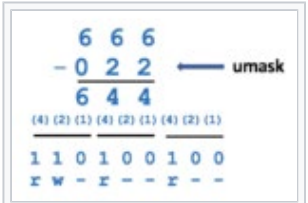
Using octal numbers to represent setting file permissions.

Setting Permissions for Newly-Created Directories and Regular Files (umask):

The **umask** command is used to set the permissions of newly-created directories and regular files.



Setting **umask** for newly-created **directories**.



Setting **umask** for newly-created **regular files**

Issuing the **umask** command without arguments will display the current umask value. Refer to the diagrams on the right-side to set the umask value for directories and regular files. Setting the umask value (for example umask 022) only takes effect for the current shell session unless the umask command is contained in a start-up file (e.g. **.profile**, **.bash_profile**, or **.bashrc**).

INVESTIGATION 1: NUMBERING CONVERSIONS

ATTENTION: This online tutorial will be required to be completed by **Friday in week 5 by midnight** to obtain a grade of **2%** towards this course

For this investigation, we will NOT be logged into our Matrix account, but it is recommended to have an **MS Word document**
open to manually perform numbering conversions.

NOTE: It is essential that you learn how to manually perform numbering conversions since you will NOT be permitted to perform quizzes, midterm, or your final exam with a computer or a calculator. Learning to quickly perform manual numbering conversions will make IT professional more productive such as setting permissions, designing computer networks, or selecting complex colors when developing webpages.

You will now get practice performing numbering conversions.

Only use a calculator to check your numbering conversion **AFTER** you have performed the operation **manually**.

Perform the Following Steps:

1. Let's convert the following binary number **10111110** to a **decimal number**.

   NOTE: It is important to learn and **memorize** the **correct methods** to perform the proper numbering conversion method (i.e. view **method 1** above (drawing the L's).

2. Write the manual conversion either in your MS Word document. 190

3. Use a **calculator** to check your work. In MS Windows, you can set the calculator to Programming mode by making the selection to **binary**, enter the binary number **10111110** and view the decimal equivalent.

   Did you get the correct answer? If not, retry the method and check to see what you did wrong. Yes

4. Perform a manual conversion of the **decimal number 55** to a **binary number**.
   What method (displayed above) will you use? Use a calculator to check your work.
   Method 2 00110111
5. Perform a manual conversion of the **octal number 461** to a **binary number**.
   What method (displayed above) will you use? Use a calculator to check your work.
   Method 3 100110001
6. Perform a manual conversion of the **binary number 11110001** to a **hexadecimal number**.
   What method (displayed above) will you use? Use a calculator to check your work.
   Method 4 F1
7. Perform a manual conversion of the **hexadecimal number ABC** to a **binary number**.
   What method (displayed above) will you use? Use a calculator to check your work.
   Method 4 101010111100
8. Perform a manual conversion of the **binary number 10101111** to an **octal number**.
   What method (displayed above) will you use? Use a calculator to check your work.
   Method 3 257
9. Perform a manual conversion of the same **binary number 10101111** to a **hexadecimal number**. Method 4 AF
   What method (displayed above) will you use? Use a calculator to check your work.

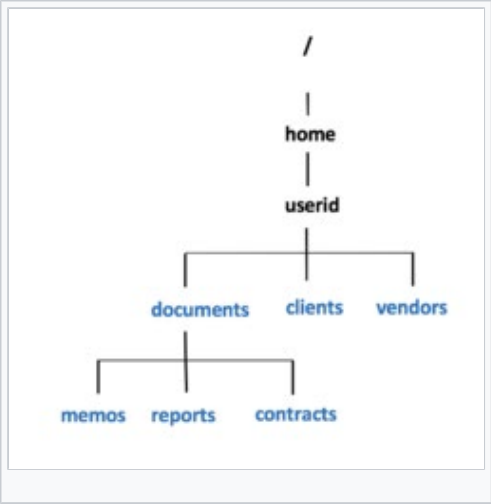
10. Perform a manual conversion of the **octal number 5636** to a **hexadecimal number**.
    What method (displayed above) will you use? Use a calculator to check your work.

    Method 5 B9E
11. Perform a manual conversion of the **hexadecimal number D68** to an **octal number**.
    What method (displayed above) will you use? Use a calculator to check your work.

    Method 5 6550
12. When you have performed all of the numbering conversions above, then you can proceed to the next INVESTIGATION.

INVESTIGATION 2: FILE PERMISSIONS

In this investigation, you will get experience using the **chmod** command to **change permissions** for existing files and the using **umask** command to automatically set permissions for newly-created files.



Perform the Following Steps:

1. **Login** to your matrix account and issue a command to **confirm** you are located in your **home** directory.

2. Issue a single Linux command to create the following directory structure displayed in the diagram to the right.

   NOTE: You will now run a shell script to confirm that you properly created that directory structure
   in your *Matrix* account.

3. Issue the following Linux command to run a checking script:
   `~uli101/week4-check-1`

4. If you encounter errors, make corrections and **re-run** the checking script until you receive a congratulations message, then you can proceed.

5. Issue Linux commands to create **empty files** for each of those newly created **directories** as shown in diagram to the right:

Memos	Reports	Contracts	clients	vendors
memo1.txt memo2.txt memo3.txt	report1.txt report2.txt	contract1.txt contract2.txt contract3.txt	linux.txt unix.txt	seneca.txt

   NOTE: You will now run another shell script to confirm that you properly created those

empty files within those specified directories.

6. Issue the following Linux command to run a checking script:
   ~uli101/week4-check-2

7. If you encounter errors, make corrections and **re-run** the checking script until you receive a congratulations message, then continue the remaining steps.

Let's get practice **viewing permissions**, **changing permissions**, and automatically **setting permissions for newly created files**.

```
/home/twwong9/clients:
total 0
-rw-r--r-- 1 twwong9 users 0 Feb 10 15:55 linux.txt
-rw-r--r-- 1 twwong9 users 0 Feb 10 15:55 unix.txt
/home/twwong9/documents:
total 0
drwxr-xr-x 2 twwong9 users 69 Feb 10 15:54 contracts
drwxr-xr-x 2 twwong9 users 57 Feb 10 15:52 memos
drwxr-xr-x 2 twwong9 users 44 Feb 10 15:53 reports
/home/twwong9/documents/contracts:
total 0
-rw-r--r-- 1 twwong9 users 0 Feb 10 15:54 contract1.txt
-rw-r--r-- 1 twwong9 users 0 Feb 10 15:54 contract2.txt
-rw-r--r-- 1 twwong9 users 0 Feb 10 15:54 contract3.txt
/home/twwong9/documents/memos:
total 0
-rw-r--r-- 1 twwong9 users 0 Feb 10 15:52 memo1.txt
-rw-r--r-- 1 twwong9 users 0 Feb 10 15:52 memo2.txt
-rw-r--r-- 1 twwong9 users 0 Feb 10 15:52 memo3.txt
/home/twwong9/documents/reports:
total 0
-rw-r--r-- 1 twwong9 users 0 Feb 10 15:53 report1.txt
-rw-r--r-- 1 twwong9 users 0 Feb 10 15:53 report2.txt
/home/twwong9/vendors:
total 0
-rw-r--r-- 1 twwong9 users 0 Feb 10 15:55 seneca.txt
```

```
drwxr-xr-x 2 twwong9 users 39 Feb 10 15:55 /home/twwong9/clients
drwxr-xr-x 5 twwong9 users 51 Feb 10 15:46 /home/twwong9/documents
drwxr-xr-x 2 twwong9 users 24 Feb 10 15:55 /home/twwong9/vendors
```

8. Issue the following Linux commands:
   ls -ld ~/documents ~/clients ~/vendors
   ls -lR ~/documents ~/clients ~/vendors

NOTE: You should see permissions already set for those newly created directories and regular files.
What do these permissions mean for **same group member** and **other group member** access to those directory and regular files?

9. Let's limit access to the **clients** and **vendors** directories to only yourself and same group members.
   Issue the following Linux command:
   chmod 750 ~/clients ~/vendors

```
drwxr-x--- 2 twwong9 users 39 Feb 10 15:55 /home/twwong9/clients
drwxr-xr-x 5 twwong9 users 51 Feb 10 15:46 /home/twwong9/documents
drwxr-x--- 2 twwong9 users 24 Feb 10 15:55 /home/twwong9/vendors
```

10. Issue the **ls -ld** and **ls -lR** commands (as you did in *step #8*) to confirm that the permissions for those directories have been changed.

NOTE: The **-R** option for the **chmod** command can change the file permissions recursively within a directory structure.

11. Issue the following Linux command: **chmod 750 -R ~/documents**

12. Issue the **ls -ld** command to confirm the permissions for the **~/documents**, **~/documents/memos** , **~/documents/reports**, and **~/documents/contracts** directories.

```
drwxr-x--- 5 twwong9 users 51 Feb 10 15:46 /home/twwong9/documents
```

```
drwxr-x--- 2 twwong9 users 57 Feb 10 15:52 /home/twwong9/documents/memos
drwxr-x--- 2 twwong9 users 44 Feb 10 15:53 /home/twwong9/documents/reports
drwxr-x--- 2 twwong9 users 69 Feb 10 15:54 /home/twwong9/documents/contracts
```

13. Issue the following Linux command: **ls -lR ~/documents**
    What do you noticed happened to the permissions for the regular files contained in those directories.
    Did those regular file permissions change?
    Yes, those regular file permissions also change

    We will now change permissions for regular text file contained in subdirectories of the **documents** directory to: **r w - r - - - - -**

14. Issue the following Linux commands:
    chmod 640 ~/documents/memos/memo*.txt
    chmod 640 ~/documents/reports/report*.txt
    chmod 640 ~/documents/contracts/contract*.txt

```
/home/twwong9/documents:
total 0
drwxr-x--- 2 twwong9 users 69 Feb 10 15:54 contracts
drwxr-x--- 2 twwong9 users 57 Feb 10 15:52 memos
drwxr-x--- 2 twwong9 users 44 Feb 10 15:53 reports

/home/twwong9/documents/contracts:
total 0
-rw-r----- 1 twwong9 users 0 Feb 10 15:54 contract1.txt
-rw-r----- 1 twwong9 users 0 Feb 10 15:54 contract2.txt
-rw-r----- 1 twwong9 users 0 Feb 10 15:54 contract3.txt

/home/twwong9/documents/memos:
total 0
-rw-r----- 1 twwong9 users 0 Feb 10 15:52 memo1.txt
-rw-r----- 1 twwong9 users 0 Feb 10 15:52 memo2.txt
-rw-r----- 1 twwong9 users 0 Feb 10 15:52 memo3.txt

/home/twwong9/documents/reports:
total 0
-rw-r----- 1 twwong9 users 0 Feb 10 15:53 report1.txt
-rw-r----- 1 twwong9 users 0 Feb 10 15:53 report2.txt
```

15. Issue the **`ls -lR`** command for the **~/documents** directory to confirm that those regular file permission have changed.

    Let's run a checking script to make certain you correctly set permissions for those directories and files.

16. Issue the following: **`~uli101/week4-check-3`**

17. If you encounter errors, make corrections and then re-run the checking script until you receive a congratulations message
    and then continue with this tutorial.

    Let's get some practice setting permissions to allow users to make editing changes to regular files.

```
-r--r----- 1 twwong9 users 0 Feb 10 15:52 memo1.txt
-r--r----- 1 twwong9 users 0 Feb 10 15:52 memo2.txt
-r--r----- 1 twwong9 users 0 Feb 10 15:52 memo3.txt
```

18. Issue the following Linux command: **`chmod ugo-w ~/documents/memos/memo*.txt`**

19. Use the **`ls`** command to verify that those regular file's permissions have changed.

20. Using the nano or vi text editor, open the regular file **~/documents/memos/memo1.txt** and type in some text and try to save your editing changes.
    What happened?
    W10: Warning: Changing a readonly file

21. To **abort** your editing session in **vi**: type **`:q!`** and press **ENTER**.
    To **abort** your editing changes in **nano**: type **`ctrl-x`**
    type **`n`** and then press **ENTER** when prompted to save editing changes.

22. Issue the following Linux command to add write permissions for all files in the **memos** directory
    for yourself (i.e. user): **`chmod u+w ~/documents/memos/*`**

23. Repeat steps to edit the file **~/documents/memos/memo1.txt** (as you did in *step #20*).
    Were you able to edit the file and save your editing changes?
    Yes I am

24. Issue a Linux command to view the contents of the **~/documents/memos/memo1.txt** text file that you were able to edit.
    cat ~/documents/memos/memo1.txt

25. Issue the following Linux command to view permissions for your **home** directory: **`ls -ld ~`**

    What does execute permissions mean for same group members and other group members in terms of your **home** directory?
    drwx--x--x 12 twwong9 users 319 Feb 10 16:52 /home/twwong9

26. Issue the following Linux command to create a new subdirectory: **`mkdir ~/shared`**

27. Issue the following Linux command: **`ls -ld ~/shared`**
    drwxr-xr-x 2 twwong9 users 6 Feb 10 16:56 /home/twwong9/shared
    What are the permissions for this newly-created directory?
    Can other users access the directory pathname **`~youruserid/shared`** ?
    I think they can access

28. Issue the following Linux command (without an argument): **`umask`**

    NOTE: You should see a **four-digit octal** number. Drop the leading zero on the left to obtain the **default umask value**. 022

29. Perform a **mathematical calculation** by taking the octal number **777** and subtracting the default umask value you determined in the previous step. What is the result?
    755
30. Convert that octal number result to a **binary** number. What does that represent as newly created directory permissions? 111 101 101
    Does that correspond to the permissions for the newly created **~/shared** directory?
    Yes, it corresponds
31. Repeat the calculation (like in step #28) but with a umask setting of **077** to see how this new umask setting would
    affect permissions of newly-created directories.

32. Issue the following Linux command: **`umask 077`**

33. Issue the following Linux command (without arguments): **`umask`**

    NOTE: You should notice the value **0077**. By dropping the leading zero to the left, that would provide the default **umask value of 077**.

34. Issue the following Linux command: **`mkdir ~/shared2`**

35. Issue the following Linux command: **`ls -ld ~/shared2`**

    Do the permissions for this newly created directory match the predicted permissions that you calculated in **step #30**? Yes, 700 = 111 000 000
    drwx------ 2 twwong9 users 6 Feb 10 17:04 /home/twwong9/shared2
36. Issue the following Linux command to create an empty regular file called **myfile.txt** in the **~/shared2** directory:
    `touch ~/shared2/myfile.txt`

37. Use the **`ls -l`** command to view the permissions for this newly created regular file.

    What do you notice about those permissions?
    -rw------- 1 twwong9 users 0 Feb 10 17:07 /home/twwong9/shared2/myfile.txt
    Let's run a checking script to make certain you correctly set permissions for those recently-created directories and files.

38. Issue the following: **`~uli101/week4-check-4`**

    If you encounter errors, make corrections and then re-run the checking script until you receive a congratulations message
    and then continue with this tutorial.

39. Logout of your Matrix account, and then log-back into your Matrix account.

40. Issue the following Linux command (without arguments): `umask`

    What happened? Referring to your notes, what do you need to do to make that umask value persistent? umask reset to 022

    WARNING:
    You should be extremely aware of your permissions since you may perform **marked work** for other courses on your **Matrix** server.
    You should NOT set permissions to share your work with **same group** or **other group** members (unless given **specific permissions instructions from your course professors**). If students can have access to your directories and project files, they could **copy** your work and thus make yourself and other student(s) that copied your work to be charged with **academic dishonesty**.

Complete the Review Questions sections to get additional practice.

LINUX PRACTICE QUESTIONS

The purpose of this section is to obtain extra practice to help with your quizzes, your midterm, and your final ezam.

Here is a link to the MS Word Document of ALL of the questions displayed below but with extra room to answer on the document to simulate a quiz:

https://github.com/ULI101/labs/raw/main/uli101_week4_practice.docx

Your instructor may take-up these questions during class. It is up to the student to attend classes in order to obtain the answers to the following questions. Your instructor will NOT provide these answers in any other form (eg. e-mail, etc).

Review Questions:

1. List the number of digits for the following numbering systems:
   - **Decimal**
   - **Binary**
   - **Octal**
   - **Hexadecimal**

2. Write a simple chart to show which values are represented for letter **A - F** for a hexadecimal number.
3. How many **binary** digits does 1 octal digit represent?
4. How many **binary** digits does 1 hexadecimal digit represent?
5. Use **manual numbering conversion** to complete the table displayed to the right.

Decimal	Binary	Octal	Hexadecimal
101			
	11110011		
		56	
			AC

6. Write the **chmod** command (using the *symbolic* method) to set "**pass-through**" permissions (eg. `r w x - - x - - x`) for your **home** directory using an **absolute pathname**.
   Write a Linux command to verify that permissions where set.

7. Perform a binary to octal numbering conversion for the permissions: **r w x - - x - - x**
   Write single Linux command to set "**pass-through**" permissions for your **home** directory, using the **absolute method** (i.e. octal numbers).

8. Write a single Linux command to **add read permissions** for **same group members** for the **~/tests** directory.

9. Write a single Linux command to **remove write permissions** for **same group members** and **other group members** for the **~/projects** directory. Use the *symbolic* method.

10. Write a single Linux command to set the permissions for the **~/assignments** directory to the following using the **absolute** method (i.e. octal numbers): **r w x r - x - - x**
    Show your work to perform a **binary** to **octal** conversion.
    Write the command below using octal numbers and using a relative-to-home pathname.

11. Assume that you just issued the command:
    ```
    chmod u=rwx,go=x ~/linux/content
    ```
    What would be the new permissions for the "**content**" directory?

12. Assume that you just issued the commands:
    ```
    umask 077
    mkdir mydir
    touch mydir/myfile.txt
    ```
    What would be the permissions for the newly created **directory** and **regular file**?
    (show your work)

Author: Murray Saul

License: LGPL version 3 Link: https://www.gnu.org/licenses/lgpl.html

---

Category: ULI101