OSL640: INTRODUCTION TO OPEN SOURCE SYSTEMS

WEEK 4: LESSON 1

DATA REPRESENTATION
NUMBERING CONVERSION

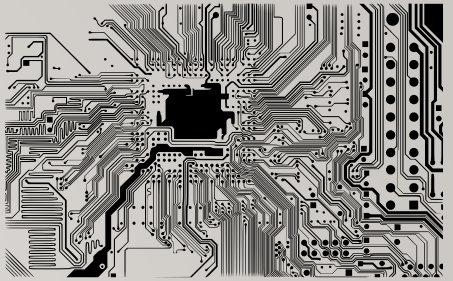
LESSON 1 TOPICS

Data Representation

- Purpose
- **Decimal**, **Binary**, **Octal**, **Hexadecimal** Numbering Systems
- Numbering Conversion Methods
- Demonstration

Perform Week 4 Tutorial

- Investigation 1
- Review Questions (Questions 1 – 5)



DATA REPRESENTATION

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 (0's and 1's) are combined in a sequence to form a **byte**. Bytes are used to represent **numbers** or **characters**.

It is the job of the computer program to understand if those bytes (series of o's and/or 1's) represent numbers or characters (eg. in **C programming**, declaring a variable with a **data type**)

Understanding how the computer stores numbers and characters can be useful when **administrating computer systems** and **creating programs** to be run on computer systems.

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

	0	1	2	3	4	5	6	7
0	NUL	DLE	space	0	@	P	`	p
1	SOH	DC1 XON	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3 XOFF	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	|
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	del

DATA REPRESENTATION

Numbering Conversion:

Computers have evolved over time. During that time, humans have interfaced with the computer by *binary* numbers, or by using **short-cuts** such as **octal** or **hexadecimal** numbers.

Computer Networking / Support Specialists and **Computer Programmers** occasionally need to convert between numbering systems:

- Converting **decimal** numbers to **binary** number for URLs (subnetting)
- Converting **decimal** numbers to **hexadecimal** numbers to format webpages (with web-safe colours)
- Converting **binary** numbers to **octal** numbers for setting file permissions in Unix/Linux

Before performing numbering conversions, we need to better understand the **decimal**, **binary**, **octal** and **hexadecimal** numbering systems.

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

	0	1	2	3	4	5	6	7
0	NUL	DLE	space	0	@	P	`	p
1	SOH	DC1 XON	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3 XOFF	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	|
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	del

DATA REPRESENTATION

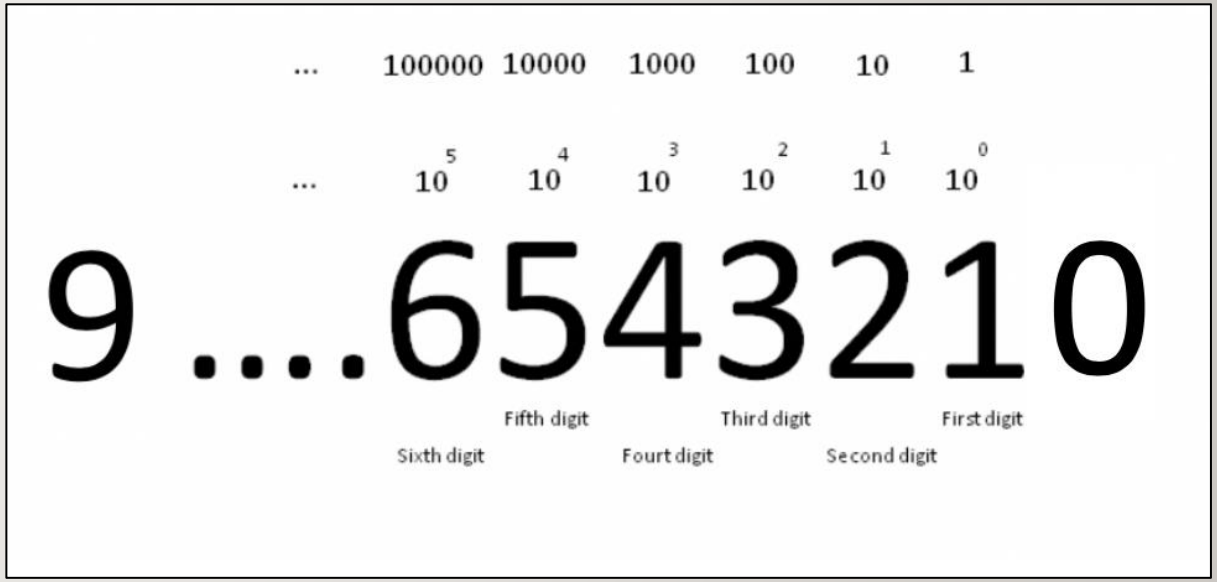
Decimal Numbering System (Humans)

The numbering system used by **humans**.

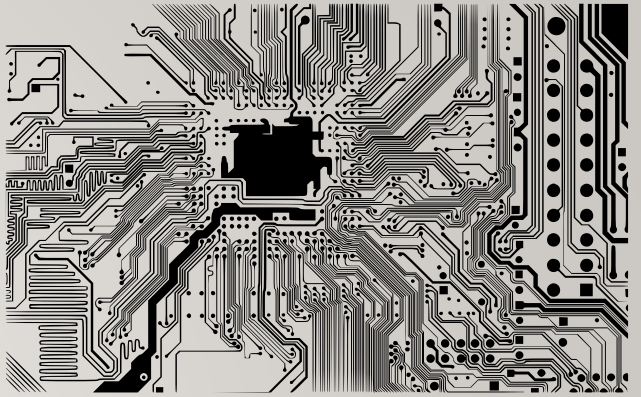
The **decimal** numbering system consists of **digits** ranging from **0** to **9**.

The fact that **humans** started counting on their **fingers** and **thumbs** most likely lead to the development of this numbering system.

The decimal numbering system is based on **sums of the power of 10** which provides a framework for mathematic calculations.



DATA REPRESENTATION

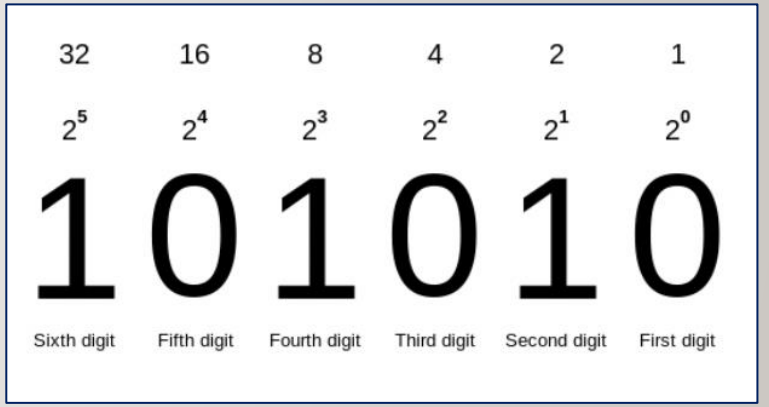


Binary Numbers
(Computers)

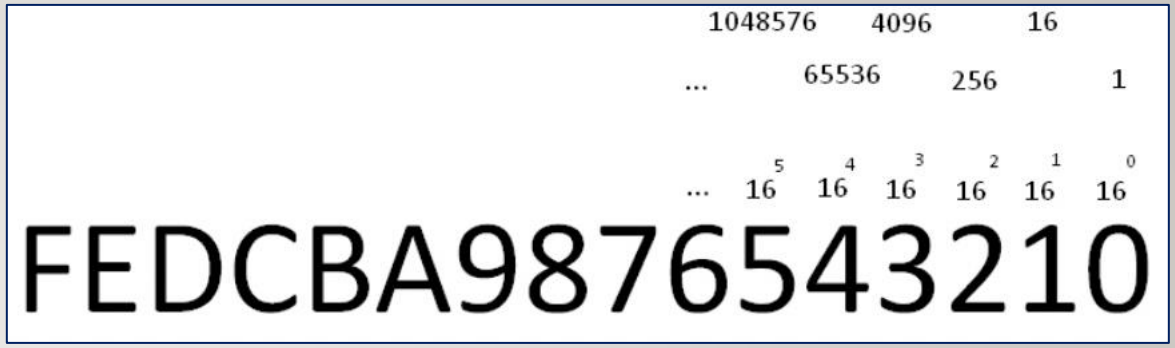
Digital computers have **circuits** which representing data in terms of voltage levels. Multiple circuits are used to represent data
(in the form of *binary* numbers).

The **binary** numbering system consists of digits ranging from **0** to **1**.
The numbering system is based on sums of the power of **2**.

Referring to the diagram to the right, the value of each decimal digit consists of the value (placeholder) multiplied by the corresponding power of 2. For example, $\mathbf{2^0}$, $\mathbf{2^1}$, $\mathbf{2^2}$, etc. which move in a **right-to-left** direction.



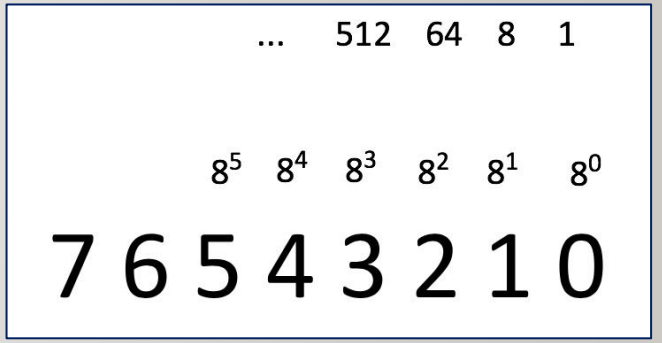
DATA REPRESENTATION



Octal / Hexadecimal Numbers (short-cuts)

The **octal** and **hexadecimal** numbering systems consist of digits ranging from **0 to 7** and ranging from **0 to F** respectively.

The **octal** and **hexadecimal** numbering system are based on sums of the power of **8** and **16** respectively. For *hexadecimal* numbers, values for **10 to 15** are represented by the characters **A to F** respectively.

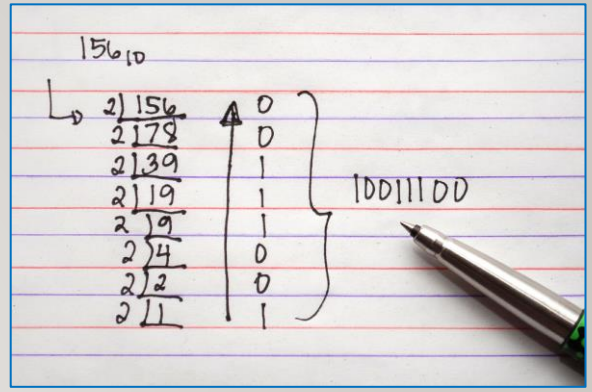


These numbering systems are useful since they are **both multiples of 2** (binary) and can be used as **short-cuts** to represent a series of binary numbers:

1 octal digit = 3 binary digits

1 hexadecimal digit = 4 binary digits).

DATA REPRESENTATION



Performing Numbering Conversion

You will learn **several numbering conversion methods** in this course:

1. **Binary** to **Decimal**
2. **Decimal** to **Binary**
3. **Octal** to **Binary** / **Binary** to **Octal**
4. **Hexadecimal** to **Binary** / **Binary** to **Hexadecimal**
5. **Octal** to **Hexadecimal** / **Hexadecimal** to **Octal**

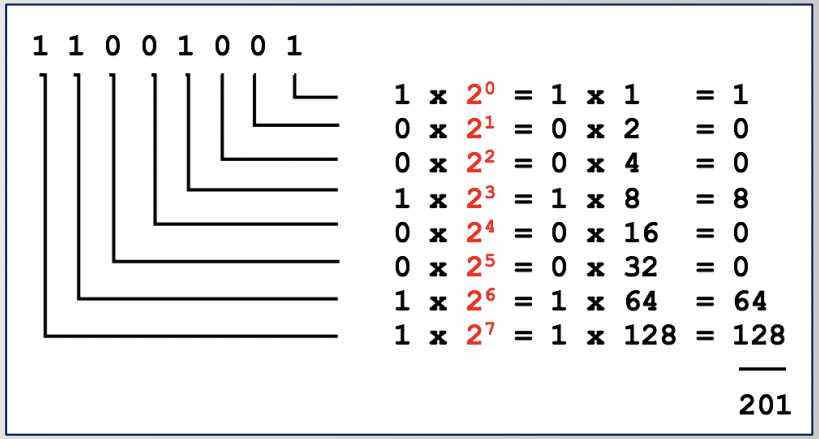
NOTE: Each of these techniques are **unique**. You will be expected not only to perform these calculations on a *quiz* / *midterm exam* / *final exam* but also **show your work** and **use the same technique show in these slides** to obtain full marks.

DATA REPRESENTATION

Numbering Conversion 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).
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*).

DATA REPRESENTATION

Instructor Demonstration

Your instructor will now demonstration how to perform a **Binary** to **Decimal** conversion

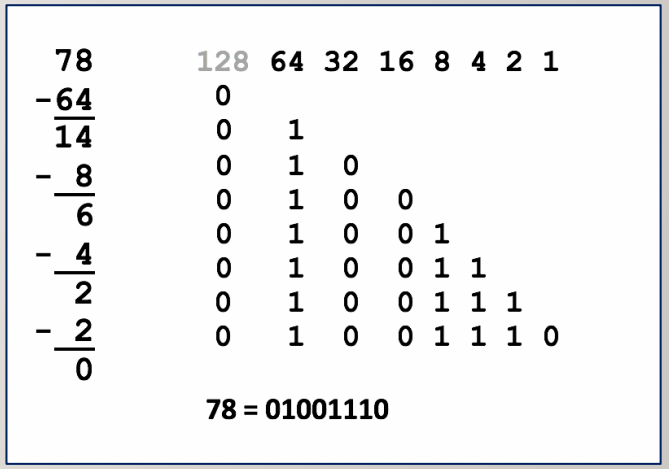
DATA REPRESENTATION

Numbering Conversion Method 2: Decimal to Binary

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

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.



DATA REPRESENTATION

Instructor Demonstration

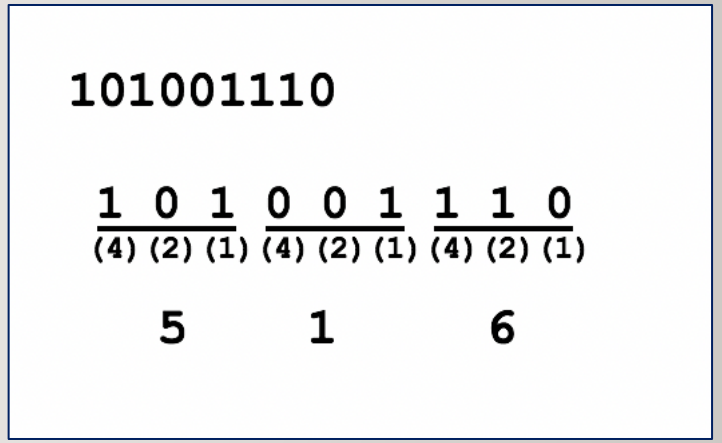
Your instructor will now demonstration how to perform a **Decimal** to **Binary** conversion

DATA REPRESENTATION

Numbering Conversion Method 3: Octal to Binary / Binary to Octal

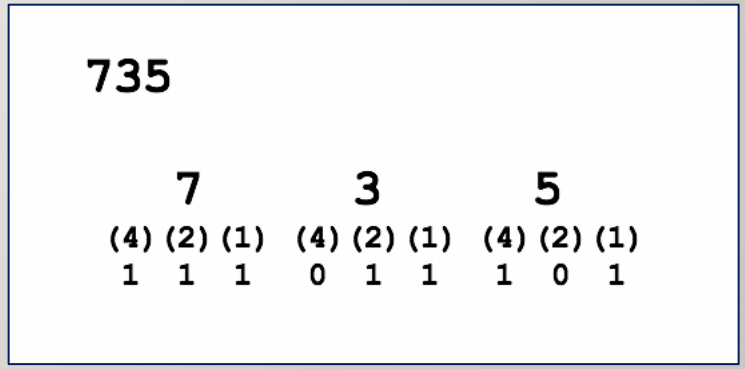
Binary to Octal

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).



DATA REPRESENTATION

Instructor Demonstration

Your instructor will now demonstration how to perform an **Octal** to **Binary** conversion and a **Binary** to **Octal** conversion.

DATA REPRESENTATION

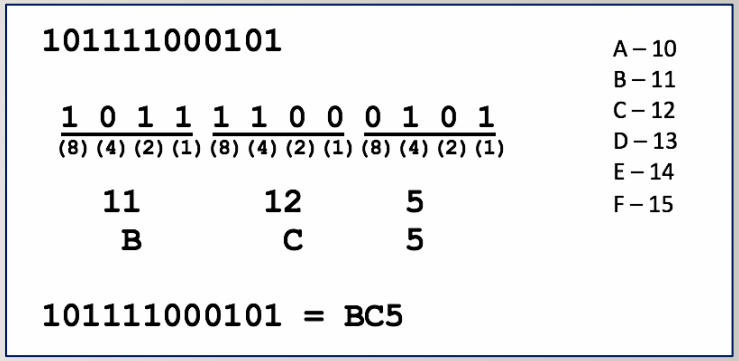
Numbering Conversion
Method 4: Hexadecimal to Binary / Binary to Hexadecimal

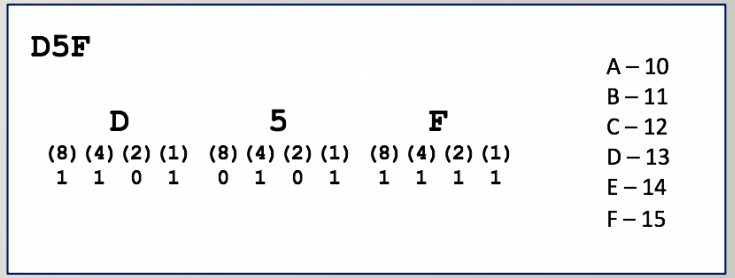
Binary to Hexadecimal

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

Hexadecimal to Binary

- **One hexadecimal number** represents **4 binary numbers**,
  so space-out the hexadecimal numbers to make space for a binary number.
- Convert letters **A** to **F** to **10** to 15 (refer to diagram of *binary to hexadecimal* conversion)
- Write **(8)(4)(2)(1)** under <u>each</u> hexadecimal number.
- 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).





DATA REPRESENTATION

Instructor Demonstration

Your instructor will now demonstration how to perform a **Hexadecimal** to **Binary** conversion and a **Binary** to **Hexadecimal** conversion.

DATA REPRESENTATION

Numbering Conversion Method 5: Octal to Hexadecimal / Hexadecimal to Octal

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

Example:

```
Octal -> binary -> Hexadecimal

Hexadecimal -> binary -> Octal
```

- 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.

DATA REPRESENTATION

Instructor Demonstration

Your instructor will now demonstration how to perform an **Octal** to **Hexadecimal** conversion and a **Hexadecimal** to **Octal** conversion.

DATA REPRESENTATION

Getting Practice

Perform the online tutorial **Tutorial 4: Data Representation / Numbering Conversions / File Permissions** (**ctrl-click** to open link):

- INVESTIGATION 1: NUMBERING CONVERSIONS
- LINUX PRACTICE QUESTIONS (Questions **1 – 5**)

ULI101: INTRODUCTION TO UNIX / LINUX AND THE INTERNET

WEEK 4: LESSON 2

FILE PERMISSIONS

LESSON 2 TOPICS

File Permissions

- Purpose
- **Directory** vs. **Regular File** Permissions
- Changing File Permissions (**chmod**)
- Setting File Permissions for Newly Created Directories and Regular Files (**umask**)
- Demonstration

Perform Week 4 Tutorial

- Investigation 2
- Review Questions (Questions 6 – 12)

FILE PERMISSIONS

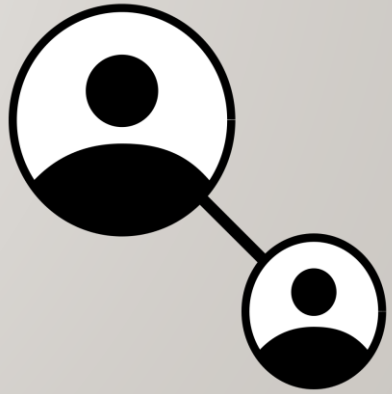
```
drwxr-xr-x 2 murray.saul users 6 Jan 19 14:06 mydir
-rw-r--r-- 1 murray.saul users 0 Jan 19 14:05 myregfile
```

File Permissions

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

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 that **same primary group** are referred to as **same group members**, and those users are do NOT belong to a particular group are referred to as **other group members**.



FILE PERMISSIONS

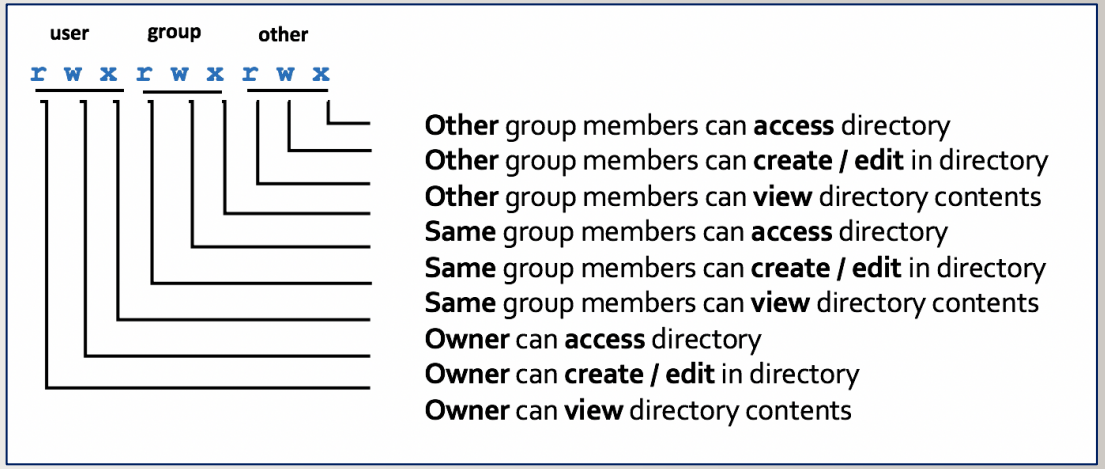
File Permissions consist of **two-layers**:

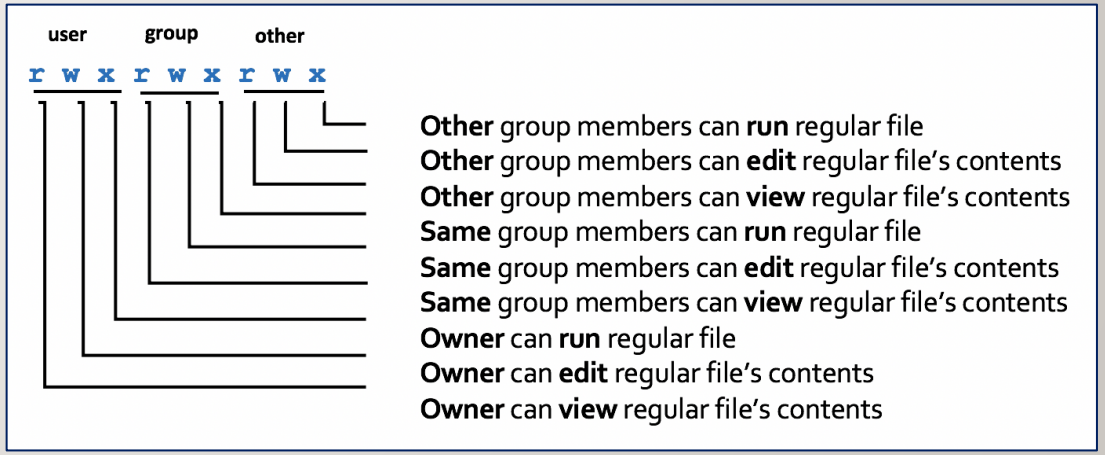
First, the permissions relating to a **directory.** Refer to the diagram on the right-side for directory permissions.

S***econd***, the permissions relating to the **regular files** contained within a directory. Refer to the diagram on bottom right-side for regular file permissions.

NOTE: *Permissions for* ***directories*** *have a different meaning than permissions for* ***regular files***.

NOTE: A symbol *dash* "**-**" indicates that the permission is **NOT** granted.





FILE PERMISSIONS

Changing File Permissions with `chmod` command: *Symbolic Method:*

The **chmod** command can use **symbols** to **add**, **remove**, and **set `rwx`** permissions for **user**, **same group members**, **other group members** or **ALL** categories:

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

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**

FILE PERMISSIONS

Instructor Demonstration

Your instructor will now demonstrate how to **add**, **remove** and **set** permissions with the **chmod** command the *Symbolic* method

FILE PERMISSIONS

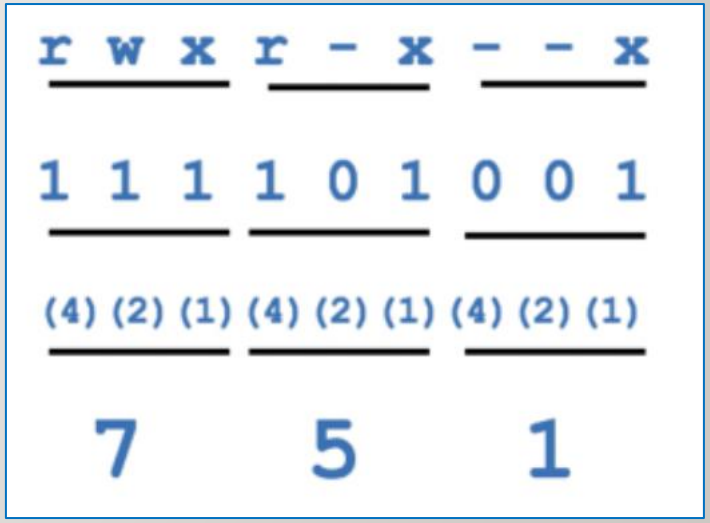
Changing File Permissions with `chmod` command:
Absolute (Octal) Method

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.

First, write **permissions** for user, group and others that you want to set**. If permission is granted, write 1 and if not granted, write 0**.

Second, perform a **binary to octal conversion**, for each group of three (user, group, other) and then issue the **chmod** command using the Absolute / Octal method.

You can only use this method to **set** file permissions (as opposed to *adding* or *removing* permissions).



FILE PERMISSIONS

Changing File Permissions with `chmod` command: *Absolute (Octal) Method*

Below is a table that displays common **chmod** commands (using the Absolute / Octal method) for common purposes.

Command	Description
`chmod 500 script.bash`	Set read and execute permissions for only the **user** for the file **script.bash** so it can be run.
`chmod 711 ~`	Set "**pass-thru**" permissions of your **home** directory.
`chmod 750 ~/shared`	Set full permissions for user, read and access permissions for some group members and no permissions for other group members for the directory **~/shared**
`chmod 555 myfile.txt`	Set read and execute permissions for the directory **myfile.txt**

FILE PERMISSIONS

Instructor Demonstration

Your instructor will now demonstrate how to **set** permissions with the **chmod** command using the *Absolute* / *Octal* method.

FILE PERMISSIONS

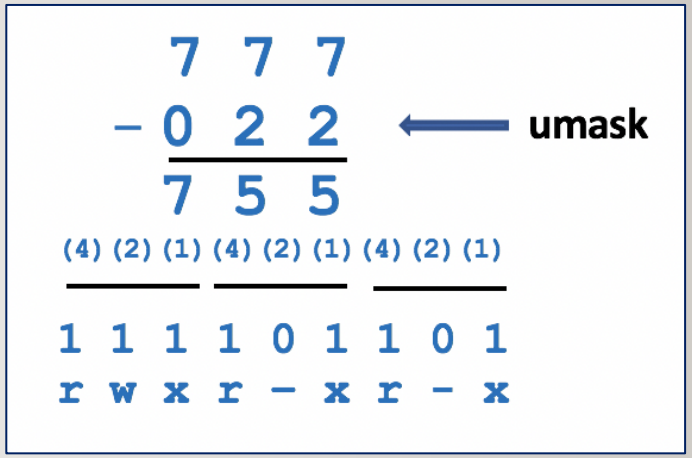
Setting Permissions for Newly-Created Directories and Regular Files (umask):

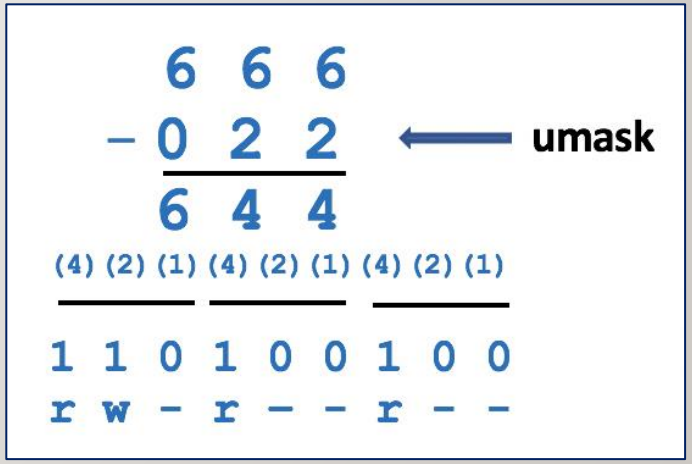
The `umask` command is used to set the permissions of **newly-created directories and regular files**. Issuing the `umask` command without arguments will display the current umask value.

The diagram on the above right shows how to calculate permissions for newly-created **directories** using the **umask** command.

The diagram on the below right shows how to calculate permissions for newly-created **regular files** using the **umask** command.

Setting the **umask** value works only in the current shell session unless the umask command is contained in a start-up file (e.g. **.profile**, **.bash_profile**, or **.bashrc**). Start-up files are discussed at the end of this course.





FILE PERMISSIONS

Instructor Demonstration

Your instructor will now demonstrate how to **set / confirm** permissions of newly-created directories and regular files using the **umask** command.

FILE PERMISSIONS

Getting Practice

Perform the online tutorial **Tutorial 4: Data Representation / Numbering Conversions / File Permissions** (**ctrl-click** to open link):

- INVESTIGATION 2: FILE PERMISSIONS
- LINUX PRACTICE QUESTIONS (Questions 6 – 12)