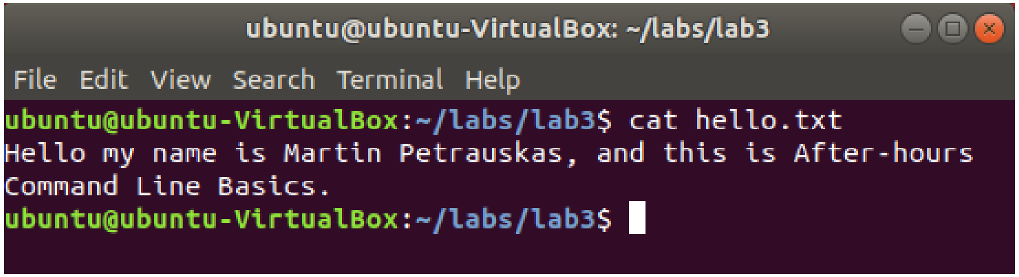
File Permissions

What are files?

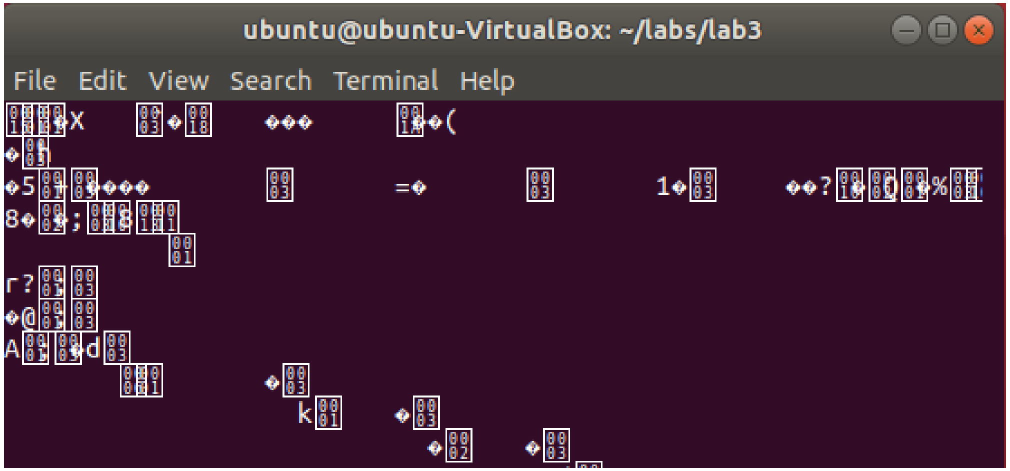
- Files are bytes of data written on some computer resource
- To see the contents of a file, use the **cat** command
- The syntax is **cat [filename]**

Text Files vs. Binary Files

- There are two types of files
  - Text: contain characters that can be read by humans
  - Binary: contains characters and symbols that can only be read by computers


ubuntu@ubuntu-VirtualBox: ~/labs/lab3
File Edit View Search Terminal Help

```
ubuntu@ubuntu-VirtualBox:~/labs/lab3$ cat hello.txt
Hello my name is Martin Petrauskas, and this is After-hours
Command Line Basics.
ubuntu@ubuntu-VirtualBox:~/labs/lab3$ 
```

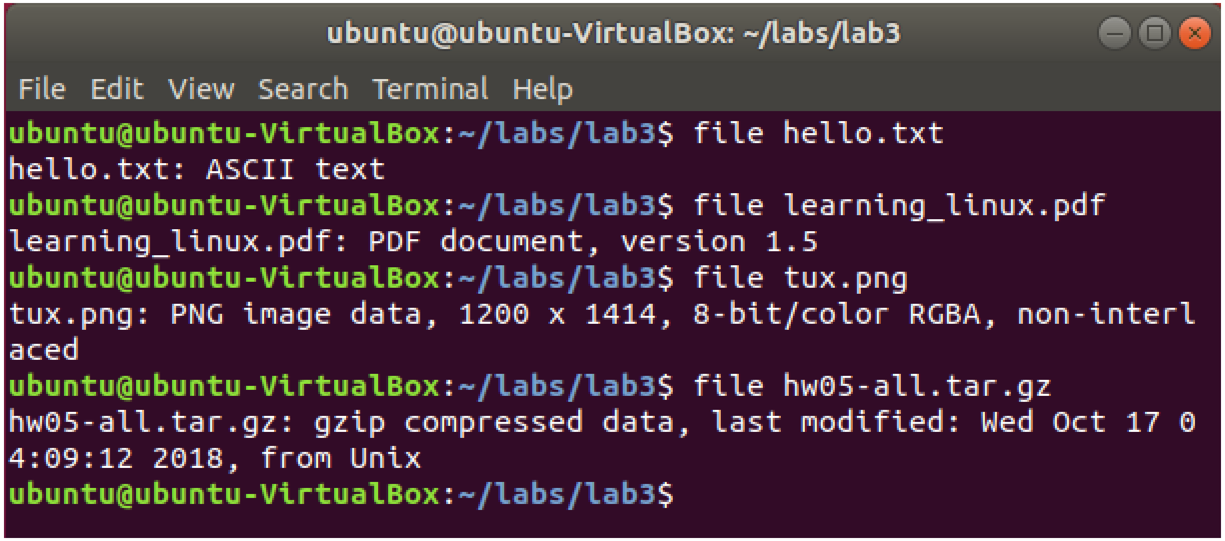

ubuntu@ubuntu-VirtualBox: ~/labs/lab3
File Edit View Search Terminal Help

Output of: **cat hello.txt**

Output of: **cat ssort**

Determining File Type

- Use the **file** command to determine what the contents of a file are
- The syntax is **file [filename]**

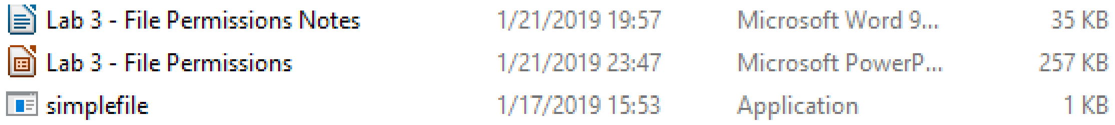


ubuntu@ubuntu-VirtualBox: ~/labs/lab3
File Edit View Search Terminal Help

```
ubuntu@ubuntu-VirtualBox:~/labs/lab3$ file hello.txt
hello.txt: ASCII text
ubuntu@ubuntu-VirtualBox:~/labs/lab3$ file learning_linux.pdf
learning_linux.pdf: PDF document, version 1.5
ubuntu@ubuntu-VirtualBox:~/labs/lab3$ file tux.png
tux.png: PNG image data, 1200 x 1414, 8-bit/color RGBA, non-interl
aced
ubuntu@ubuntu-VirtualBox:~/labs/lab3$ file hw05-all.tar.gz
hw05-all.tar.gz: gzip compressed data, last modified: Wed Oct 17 0
4:09:12 2018, from Unix
ubuntu@ubuntu-VirtualBox:~/labs/lab3$
```

File Extensions Don't Mean Anything

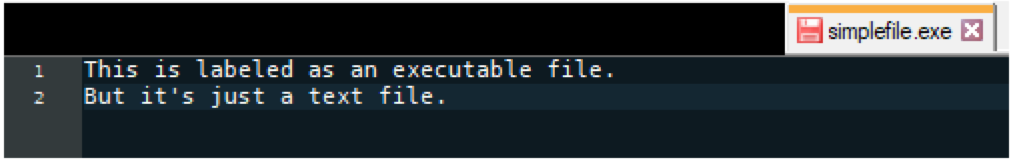
- File extensions are just a convenience
- They assist the user interface to decide which program should open a file
- However, file extensions can be incorrect



Lab 3 - File Permissions Notes	1/21/2019 19:57	Microsoft Word 9...	35 KB
Lab 3 - File Permissions	1/21/2019 23:47	Microsoft PowerP...	257 KB
simplefile	1/17/2019 15:53	Application	1 KB

This isn't an application

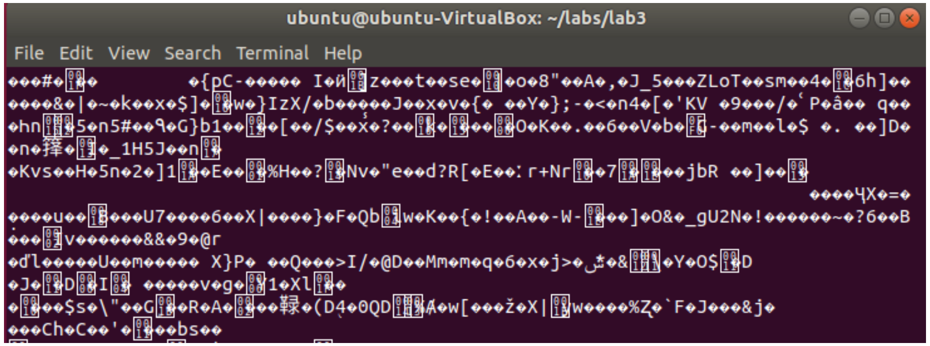
It's just a text file.



```
This is labeled as an executable file.
But it's just a text file.
```

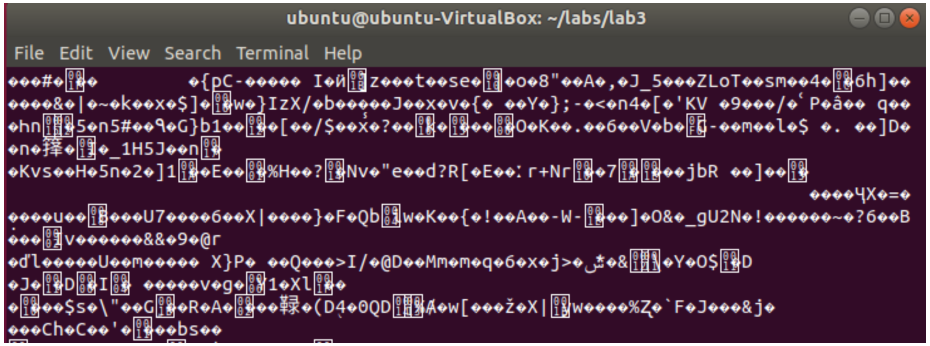
File Extensions Don't Mean Anything

- Here are two files with the same content and different file extensions



ubuntu@ubuntu-VirtualBox: ~/labs/lab3
File Edit View Search Terminal Help

```
���#�□�        �{pC-����� I�Й□z���t��se�□�o�8"��A�,�J_5���ZLoT��sm��4�□�6h]��
����&�|�~�k��x�$]�□�w�}IzX/�b�����J��x�v�{� ��Y�};-�<�n4�[�'KV �9���/�ʿP�â�� q��
�hn□□�5�n5#��٩�G}b1��□��[��/$��x�?��□k�□���□O�K��.��6��V�b�□-��m��l�$ �. ��]D�
�n�擇�□�_1H5J��n□�
�Kvs��H�5n�2�]1□��E��□�%H��?□�Nv�"e��d?R[�E��: r+Nr□��7□□���jbR ��]��□�
                                                                      ����ԿX�=�
����u��□���U7����6��X|����}�F�Qb□w�K��{�!��A��-W-□���]�O&�_gU2N�!������~�?6��B
���□v�������&&�9�@r
�ďl�����U��m������ X}P� ��Q���>I/�@D��Mm�m�q�6�x�j>�ش�&□□�Y�O$□�D
�J�□�D□�I□� �����v�g�□1�Xl□��
�□���$s�\"��G□��R�A�□���籙�(D4�0QD□□%A�w[���ž�X|□w�����%Ȥ�`F�J���&j�
���Ch�C��'�□���bs��
```



ubuntu@ubuntu-VirtualBox: ~/labs/lab3
File Edit View Search Terminal Help

```
���#�□�        �{pC-����� I�Й□z���t��se�□�o�8"��A�,�J_5���ZLoT��sm��4�□�6h]��
����&�|�~�k��x�$]�□�w�}IzX/�b�����J��x�v�{� ��Y�};-�<�n4�[�'KV �9���/�ʿP�â�� q��
�hn□□�5�n5#��٩�G}b1��□��[��/$��x�?��□k�□���□O�K��.��6��V�b�□-��m��l�$ �. ��]D�
�n�擇�□�_1H5J��n□�
�Kvs��H�5n�2�]1□��E��□�%H��?□�Nv�"e��d?R[�E��: r+Nr□��7□□���jbR ��]��□�
                                                                      ����ԿX�=�
����u��□���U7����6��X|����}�F�Qb□w�K��{�!��A��-W-□���]�O&�_gU2N�!������~�?6��B
���□v�������&&�9�@r
�ďl�����U��m������ X}P� ��Q���>I/�@D��Mm�m�q�6�x�j>�ش�&□□�Y�O$□�D
�J�□�D□�I□� �����v�g�□1�Xl□��
�□���$s�\"��G□��R�A�□���籙�(D4�0QD□□%A�w[���ž�X|□w�����%Ȥ�`F�J���&j�
���Ch�C��'�□���bs��
```

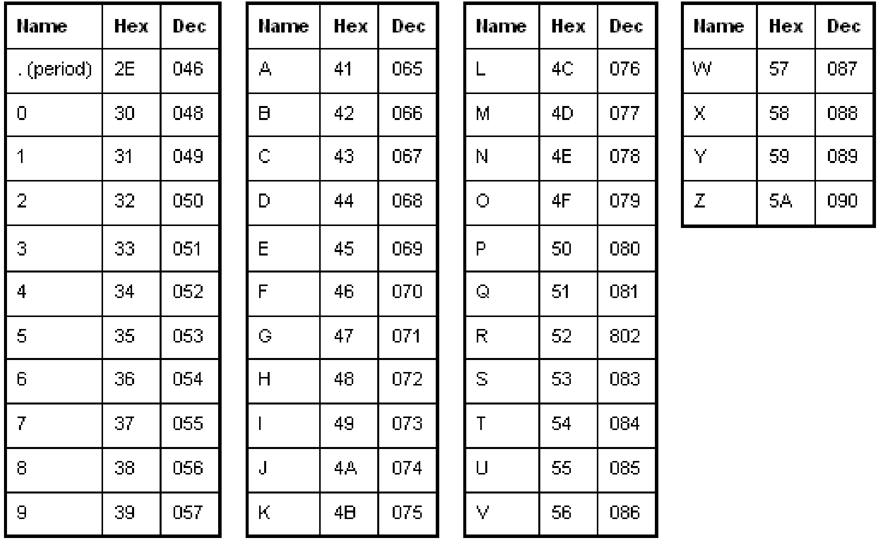
Output of **cat learning_linux.pdf**

Output of **cat learning_linux.txt**

The output contains the same information, so the extension doesn't matter

Character Encoding – ASCII

- Way of transforming regular characters into binary for computers to read
- ASCII (American Standard Code for Information Interchange)
- Allows for 128 different characters to be encoded



Name	Hex	Dec
. (period)	2E	046
0	30	048
1	31	049
2	32	050
3	33	051
4	34	052
5	35	053
6	36	054
7	37	055
8	38	056
9	39	057

Name	Hex	Dec
A	41	065
B	42	066
C	43	067
D	44	068
E	45	069
F	46	070
G	47	071
H	48	072
I	49	073
J	4A	074
K	4B	075

Name	Hex	Dec
L	4C	076
M	4D	077
N	4E	078
O	4F	079
P	50	080
Q	51	081
R	52	802
S	53	083
T	54	084
U	55	085
V	56	086

Name	Hex	Dec
W	57	087
X	58	088
Y	59	089
Z	5A	090

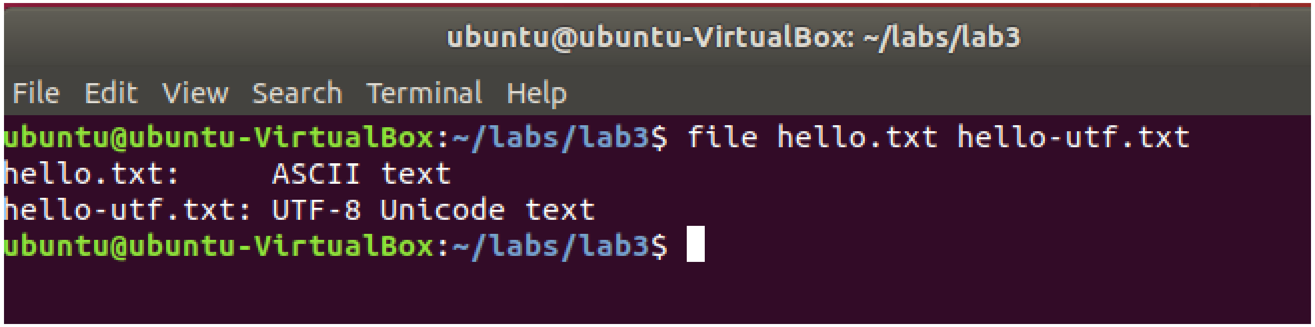
Part of an ASCII Table

Character Encoding – UTF

- Unicode Transformation Format
- Allows for over 1 million different characters to be encoded
- Unfortunately, emojis can now be encoded with UTF
- UTF is backwards compatible with ASCII

Character Encoding Example

- Operating systems will distinguish between text files depending on their encoding


ubuntu@ubuntu-VirtualBox: ~/labs/lab3
File Edit View Search Terminal Help

```
ubuntu@ubuntu-VirtualBox:~/labs/lab3$ file hello.txt hello-utf.txt
hello.txt:     ASCII text
hello-utf.txt: UTF-8 Unicode text
ubuntu@ubuntu-VirtualBox:~/labs/lab3$ 
```

LISTING FILE ATTRIBUTES

- ls –l provides attributes like
  - Permissions
  - Links
  - Owner
  - Group owner
  - Size
  - Date
  - File name

LISTING FILE ATTRIBUTES

```
$ ls –l
total 72
-rw-r--r--   1  kumar  metal  19514  may  10  13:45 chap01
-rw-r--r--   1  kumar  metal   4174  may  10  15:01 chap02
-rw-rw-rw-   1  kumar  metal     84  feb  12  12:30 dept.lst
-rw-r--r--   1  kumar  metal   9156  mar  12  1999 genie.sh
drwxr-xr-x   2  kumar  metal    512  may   9  10:31 helpdir
drwxr-xr-x   2  kumar  metal    512  may   9  09:57 progs
```

A program in detail

When we type:

```
ls -l /usr/bin/top
```

We'll see:

```
-rwxr-xr-x 1 root root 68524 2011-12-19 07:18 /usr/bin/top
```

What does all this mean?

```
-r-xr-xr-x  1   root     root     68524   2011-12-19 07:18 /usr/bin/top

---------- --- -------  -------  -------- ------------    -------------
    |       |     |        |         |         |                 |
    |       |     |        |         |         |             File Name
    |       |     |        |         |         |
    |       |     |        |         |         +---  Modification Time/Date
    |       |     |        |         |
    |       |     |        |         +-------------   Size (in bytes
    |       |     |        |
    |       |     |        +-----------------------        Group
    |       |     |
    |       |     +--------------------------------        Owner
    |       |
    |       +--------------------------------------  “link count”
    |
    +----------------------------------------------   File Permissions
```

Group

The name of the group that has permissions in addition to the file's owner.

Owner

The name of the user who owns the file.

File Permissions

The first character is the type of file. A "-" indicates a regular (ordinary) file. A "d” indicate a directory. Second set of 3 characters represent the read, write, and execution rights of the file's owner. Next 3 represent the rights of the file's group, and the final 3 represent the rights granted to everybody else.

(Example modified from http://www.linuxcommand.org/lts0030.php)

Some special cases

When looking at the output from “`ls -l`” in the first column you might see:

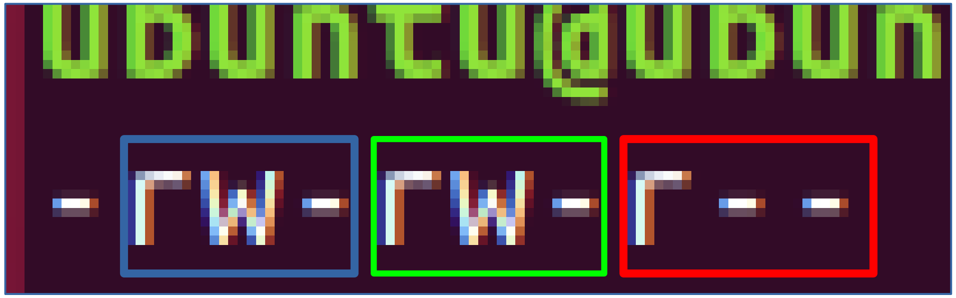
```
d = directory
- = regular file
l = symbolic link
s = Unix domain socket
p = named pipe
c = character device file
b = block device file
```

File Permissions – Background Info

- Unix systems have three levels of permissions:
  - **R**ead – user can view file contents (4)
  - **W**rite – user can edit file (2)
  - e**X**ecute – user can run file as a program or script (1)
- Users are split into three categories for permissions:
  - User/owner – user who created the file (u)
  - Group – group of users (g)
  - Other – all the other users (o)

Reading File Permissions

- How to read file permissions:



```
ubuntu@ubun
-rw-rw-r--
```

First character:
“-” (file) or
“d” (directory)

File Permissions for other users (o)

File Permissions for the user (u)

File Permissions for the group (g)

FILE OWNERSHIP

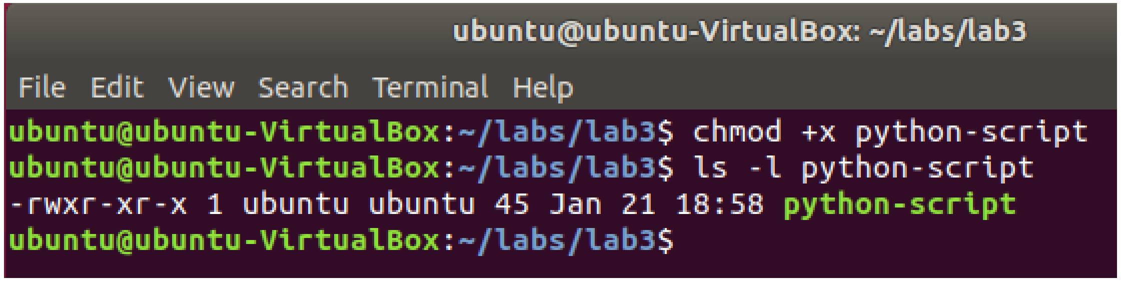
- When you create a file, you become its owner (third column)
- Group owner of the file (fourth column)
- Several users may belong to a single group, but the privileges of the group are set by the owner of the file and not by the group members

File Permissions (Cont.)

Category	operation	permission
u - user	+ assign	r - read
g - group	- remove	w - write
o - others	= absolute	x - execute
a - all (ugo)		

Changing File Permissions – Basic

- Use the **chmod** command to change file permissions
- The syntax is **chmod [mode] [filename]**


ubuntu@ubuntu-VirtualBox: ~/labs/lab3
File Edit View Search Terminal Help

```
ubuntu@ubuntu-VirtualBox:~/labs/lab3$ chmod +x python-script
ubuntu@ubuntu-VirtualBox:~/labs/lab3$ ls -l python-script
-rwxr-xr-x 1 ubuntu ubuntu 45 Jan 21 18:58 python-script
ubuntu@ubuntu-VirtualBox:~/labs/lab3$
```

Execute permissions are given to all users

File permissions (Cont.)

There are two ways to set permissions when using the chmod command:

Symbolic mode:

testfile has permissions of `-r--r--r--`

```
                              U  G  O*
$ chmod g+x testfile     ==> -r--r-xr--
$ chmod u+wx testfile    ==> -rwxr-xr--
$ chmod ug-x testfile    ==> -rw--r--r--
U=user, G=group, O=other (world)
```

File permissions cont.

Absolute mode:

We use octal (base eight) values represented like this:

Letter	Permission	Value
R	read	4
W	write	2
X	execute	1
-	none	0

For each column, User, Group or Other you can set values from 0 to 7. Here is what each means:

0= **`---`**	1= **`--x`**	2= **`-w-`**	3= **`-wx`**
4= **`r--`**	5= **`r-x`**	6= **`rw-`**	7= **`rwx`**

File permissions cont.

Numeric mode cont:

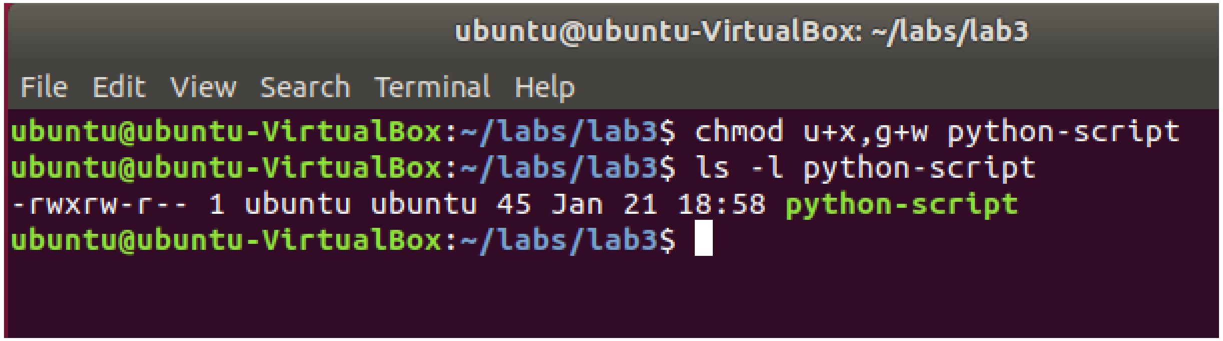
Example index.html file with typical permission values:

```
$ chmod 755 index.html
$ ls -l index.html
-rwxr-xr-x  1 root  wheel  0 May 24 06:20 index.html

$ chmod 644 index.html
$ ls -l index.html
-rw-r--r--  1 root  wheel  0 May 24 06:20 index.html
```

Changing File Permissions – Advanced

- **chmod u+x,g+w python-script**



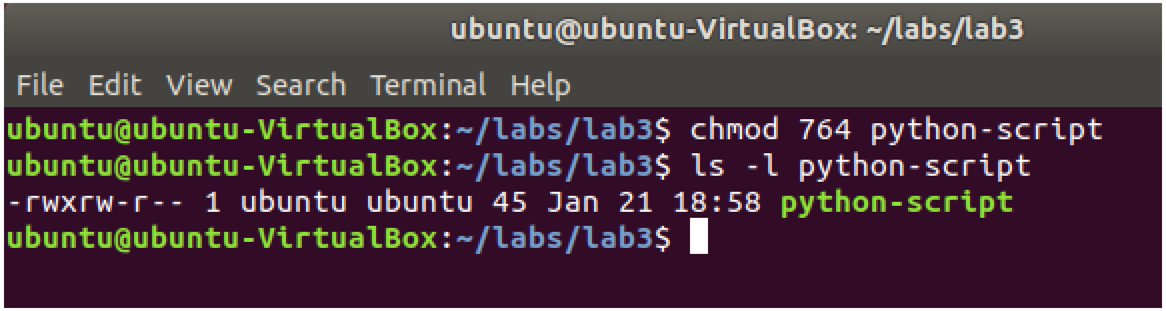
ubuntu@ubuntu-VirtualBox: ~/labs/lab3
File Edit View Search Terminal Help

```
ubuntu@ubuntu-VirtualBox:~/labs/lab3$ chmod u+x,g+w python-script
ubuntu@ubuntu-VirtualBox:~/labs/lab3$ ls -l python-script
-rwxrw-r-- 1 ubuntu ubuntu 45 Jan 21 18:58 python-script
ubuntu@ubuntu-VirtualBox:~/labs/lab3$
```

Give execute permissions to the owner and write permissions to the group

Changing File Permissions – Octal

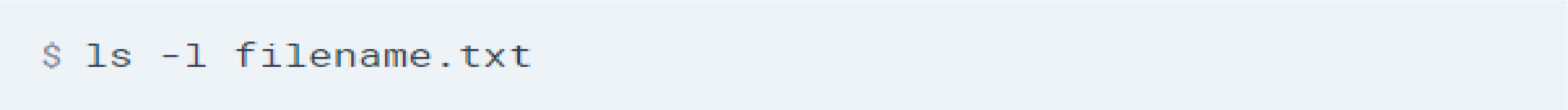
- We can use the octal number system to encode file permissions in numbers
- **chmod 764 python-script**
- Same thing as **chmod u+x,g+w python-script**
- Useful website: https://chmod-calculator.com/


ubuntu@ubuntu-VirtualBox: ~/labs/lab3
File Edit View Search Terminal Help

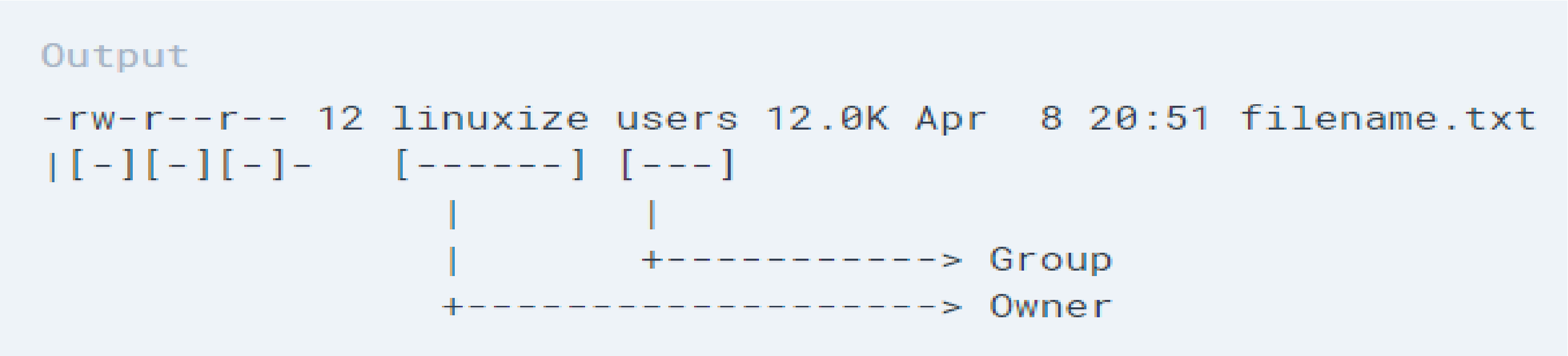
```
ubuntu@ubuntu-VirtualBox:~/labs/lab3$ chmod 764 python-script
ubuntu@ubuntu-VirtualBox:~/labs/lab3$ ls -l python-script
-rwxrw-r-- 1 ubuntu ubuntu 45 Jan 21 18:58 python-script
ubuntu@ubuntu-VirtualBox:~/labs/lab3$
```

7 is for rwx, 6 is rw-, and 4 is r--.

Changing File Ownership



```
$ ls -l filename.txt
```



```
Output
-rw-r--r-- 12 linuxize users 12.0K Apr  8 20:51 filename.txt
|[-][-][-]-   [------] [---]
                |       |
                |       +-----------> Group
                +-------------------> Owner
```

- chown linuxize file1 ...
- chown 1000 file1 # by user id

- Change the Owner and Group of a File
  - chown -R username:group directory
  - chown linuxize:users file1

Changing Group ownership

- Chgrp [OPTIONS] GROUP FILE..
- chgrp -R www-data /var/www
- chgrp +1000 filename
  - Adding new group with GID of 1000