Chapter 2 - Complimentary Linux practice booklet

About this document:

Welcome to Chapter 2 of the complimentary Linux practice book for the Apress book, "Introduction to Ansible Network Automation: The Practical Primer". This document has been created by the authors as an additional resource to the book, but it is not a part of the book itself. The purpose of this document is to provide readers with extra practice materials for Linux commands that are essential for network automation.

Chapter 2 is based on the exercises in this document and approximately 65% of the commands used in this practice booklet were utilized to form Chapter 2. Therefore, readers who are interested in learning more essential Linux commands will find additional practice materials helpful in their Linux learning journey.

If you have any questions or issues during the practice, please feel free to refer to Google, ChatGPT, Google Bar, BingGPT, Blogs, books, video trainings or any other Linux study materials that you prefer. We encourage you to experiment and explore various Linux operating systems and hope that this document will help you develop the necessary skills for network automation.

Version:	1.0
Created:	1/Apr/2023
Last updated:	N/A

What's required?

Linux OS:	**Fedora** (RHEL/CentOS) & PoPOS (Debian/Ubuntu)
Internet connection:	Yes

Contents

Linux Essential Commands 100

Exercise 1: cat & tac

time: 15 mins
==

#1-1 make a new directory ex1, change to new directory and list items.

```
localhost:~# pwd
/root
localhost:~# ls
bench.py    hello.c    hello.js    readme.txt  vi_ex
localhost:~# mkdir ex1
localhost:~# cd ex1
localhost:~/ex1# ls -lh
total 0
```

#1-2 use 'cal' to check the calendar

```
localhost:~/ex1# cal
     March 2022
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
```

#1-3 use 'clear' or [Ctrl+L] to clear your screen

```
pop-os:~/d3$ clear
pop-os:~/d3$ Ctrl+L
```

#1-4 Use the greater sign (>) to save calendar output to a file, a1 and check the content of the file using both 'cat', and 'more' (optionally, 'less')

```
localhost:~/ex1# cal > a1
localhost:~/ex1# cat a1
     March 2022
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

localhost:~/ex1# more a1
     March 2022
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
```

#1-5 Use the 'date' command, save the output to an a2 file, use 'cat' to display the content, and use 'ls' and 'ls -lh' to list the details of the file.

```
localhost:~/ex1# date
Mon Mar  7 16:40:58 UTC 2022
localhost:~/ex1# date > a2
localhost:~/ex1# cat a2
Mon Mar  7 16:41:12 UTC 2022
localhost:~/ex1# ls
a1  a2
localhost:~/ex1# ls -lh
total 8K
-rw-r--r--    1 root     root         168 Mar  7 16:34 a1
-rw-r--r--    1 root     root          29 Mar  7 16:41 a2
```

#1-6 Use 'cat' with options, -n & -b to view a1

```
localhost:~/ex1# cat a1
     March 2022
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

localhost:~/ex1# cat -n a1
     1       March 2022
     2  Su Mo Tu We Th Fr Sa
     3         1  2  3  4  5
     4   6  7  8  9 10 11 12
     5  13 14 15 16 17 18 19
     6  20 21 22 23 24 25 26
     7  27 28 29 30 31
     8
localhost:~/ex1# cat -b a1
     1       March 2022
     2  Su Mo Tu We Th Fr Sa
     3         1  2  3  4  5
     4   6  7  8  9 10 11 12
     5  13 14 15 16 17 18 19
     6  20 21 22 23 24 25 26
     7  27 28 29 30 31
     8
```

#1-7 Use vi or nano text editor to add some spacings on the a1 file, now check the difference between -n and -b

```
localhost:~/ex1# cat a1
     March 2022

Su Mo Tu We Th Fr Sa

       1  2  3  4  5

 6  7  8  9 10 11 12

13 14 15 16 17 18 19

20 21 22 23 24 25 26

27 28 29 30 31

localhost:~/ex1# cat -n a1
     1       March 2022
     2
     3  Su Mo Tu We Th Fr Sa
     4
     5         1  2  3  4  5
     6
     7   6  7  8  9 10 11 12
     8
     9  13 14 15 16 17 18 19
    10
    11  20 21 22 23 24 25 26
    12
    13  27 28 29 30 31
    14

localhost:~/ex1# cat -b a1
     1       March 2022

     2  Su Mo Tu We Th Fr Sa

     3         1  2  3  4  5

     4   6  7  8  9 10 11 12

     5  13 14 15 16 17 18 19

     6  20 21 22 23 24 25 26

     7  27 28 29 30 31
```

#1-8 Use a single cat command to display contents from multiple files

```
localhost:~/ex1# cal > a1
localhost:~/ex1# cat a1
     March 2022
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

localhost:~/ex1# cat a2
Mon Mar  7 16:41:12 UTC 2022
localhost:~/ex1# cat a1 a2
     March 2022
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

Mon Mar  7 16:41:12 UTC 2022
```

#1-10 Change the order of the files and create another file called a3

```
localhost:~/ex1# cat a2 a1 > a3
localhost:~/ex1# ls
a1  a2  a3
localhost:~/ex1# cat a3 -b
     1  Mon Mar  7 16:41:12 UTC 2022
     2       March 2022
     3  Su Mo Tu We Th Fr Sa
     4         1  2  3  4  5
     5   6  7  8  9 10 11 12
     6  13 14 15 16 17 18 19
     7  20 21 22 23 24 25 26
     8  27 28 29 30 31
     9
```

#1-11a Check the content of files a2 and a3

```
localhost:~/ex1# cat a2
Mon Mar  7 16:41:12 UTC 2022
localhost:~/ex1# cat a3
Mon Mar  7 16:41:12 UTC 2022
     March 2022
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
```

```
20 21 22 23 24 25 26
27 28 29 30 31
```

#1-11b Use two greater signs (>>) to append content of a3 to a2, then check a2.

```
localhost:~/ex1# cat a3 >> a2
localhost:~/ex1# cat a2
Mon Mar  7 16:41:12 UTC 2022
Mon Mar  7 16:41:12 UTC 2022
     March 2022
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
```

#1-11c repeat 1-11b and check a2 again.

```
localhost:~/ex1# cat a3 >> a2
localhost:~/ex1# cat a2
Mon Mar  7 16:41:12 UTC 2022
Mon Mar  7 16:41:12 UTC 2022
     March 2022
Su Mo Tu We Th Fr Saclea
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

Mon Mar  7 16:41:12 UTC 2022
     March 2022
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
```

#1-12a Use hostname to check your host's name, use cat to save the name to a file, a4

```
localhost:~/ex1# hostname
localhost
localhost:~/ex1# hostname > a4
localhost:~/ex1# cat a4
localhost
```

#1-12b Use '>' to add cal to a4, observe that the file has been overwritten.

```
localhost:~/ex1# cal > a4
localhost:~/ex1# cat a4
     March 2022
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
```

#1-12c Repeat 1-12a to update the content of a5 to the hostname, notice that when using >, you overwrite the content of your file.

```
localhost:~/ex1# hostname > a4
localhost:~/ex1# cat a4
localhost
```

#1-12d Use '>>' to append a calender after your hostname in file a5, notice that when using '>>', contents are appended to the next line of existing content.

```
localhost:~/ex1# cal >> a4
localhost:~/ex1# cat a4
localhost
     March 2022
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
```

#1-13a 'tac' is a word play on cat and it reverses the content of the file.
Check a5 first using 'cat a5', then use 'tac a5'. Observe the difference in output.

```
localhost:~/ex1# cat a4
localhost
     March 2022
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

localhost:~/ex1# tac a4

27 28 29 30 31
20 21 22 23 24 25 26
13 14 15 16 17 18 19
 6  7  8  9 10 11 12
       1  2  3  4  5
```

```
Su Mo Tu We Th Fr Sa
    March 2022
localhost
```

#1-13b We can also save the reversed content to a file using 'tac > '

```
localhost:~/ex1# tac a4 > a5
localhost:~/ex1# more a5

27 28 29 30 31
20 21 22 23 24 25 26
13 14 15 16 17 18 19
 6  7  8  9 10 11 12
       1  2  3  4  5
Su Mo Tu We Th Fr Sa
    March 2022
localhost
```

#1-14 What if you simply use the cat command? Use cat to print on the screen, and use [Ctrl+D] to quit.

```
localhost:~/ex1# cat
aaa
aaa
bbb
bbb
ccc
ccc
ddd
ddd
eee
eee
# [Ctrl+D] to quit.
```

#1-15a Use 'cat' to create a new file with a few lines. This could be faster than using the vi/pico/nano/gedit text editors.

```
localhost:~/ex1# cat a6
cat: can't open 'a6': No such file or directory
localhost:~/ex1# cat > a6
Quick
file
creation.
localhost:~/ex1# cat a6
Quick
file
creation.
# [Ctrl+D] to quit.
```

#1-15b Try the following. Create a quick python program using "cat > sayhello.py"

```
localhost:~/ex1# cat sayhello.py
cat: can't open 'sayhello.py': No such file or directory
localhost:~/ex1# cat > sayhello.py
name = input("What's your name? ")
print("Hello", name)
localhost:~/ex1# cat sayhello.py
name = input("What's your name?")
print("Hello", name)

localhost:~/ex1# python3 --version
Python 3.8.3
localhost:~/ex1# python3 sayhello.py
What's your name? Joe
Hello Joe
```

==

Ex1 Summary

cat, tac - Displays the content of files

Options:
-n displays content with row reference
-b displays content with row reference but skips blank lines

> Output to a file (overwrites the existing file with the same name)
>> Appends content to the next line (does not overwrite)

tac = is upside down the content of the file.

==

Exercise 2: tee & nl

time: 10 mins
==

#2-1 Use 'tee' to display the content while saving the output of the calendar to a file.

```
localhost:~/ex1# cd ..
localhost:~#
 ex2 && cd ex2
localhost:~/ex2#

localhost:~/ex2# cal
     March 2022
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

localhost:~/ex2# cal > a1
localhost:~/ex2# date
Mon Mar  7 17:24:01 UTC 2022
localhost:~/ex2# date > a2
localhost:~/ex2# tee
aaa
aaa
bbb
bbb
ccc
ccc
ddd
ddd
111
111
localhost:~/ex2# cat
xxx
xxx
yyy
yyy
zzz
zzz
999
999
localhost:~/ex2# cat a1
     March 2022
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
```

```
27 28 29 30 31

localhost:~/ex2# cat a1 > a3
localhost:~/ex2# cat a3
     March 2022
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

localhost:~/ex2# cat a1 | tee a4
     March 2022
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

localhost:~/ex2# cat a4
     March 2022
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
```

#2-2 Saving the output to a file using '>'.

```
localhost:~/ex2# cat /etc/passwd
root:x:0:0:root:/root:/bin/ash
bin:x:1:1:bin:/bin:/sbin/nologin
[...omitted for brevity]
nobody:x:65534:65534:nobody:/:/sbin/nologin
dhcp:x:100:101:dhcp:/var/lib/dhcp:/sbin/nologin
svn:x:101:102:svn:/var/svn:/sbin/nologin

localhost:~/ex2# cat /etc/passwd > a5
localhost:~/ex2# cat -n a5
     1  root:x:0:0:root:/root:/bin/ash
     2  bin:x:1:1:bin:/bin:/sbin/nologin
     3  daemon:x:2:2:daemon:/sbin:/sbin/nologin
[...omitted for brevity]
    27  nobody:x:65534:65534:nobody:/:/sbin/nologin
    28  dhcp:x:100:101:dhcp:/var/lib/dhcp:/sbin/nologin
    29  svn:x:101:102:svn:/var/svn:/sbin/nologin
```

#2-5 Add calender using the "tee -a" method

```
localhost:~/ex2# cal | tee -a a6
     March 2022
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

localhost:~/ex2# cat a6
     March 2022
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
```

#2-6 Modify a1 like this.

```
localhost:~/ex2# date | tee -a a1
Mon Mar  7 17:43:46 UTC 2022
localhost:~/ex2# cat a1
     March 2022
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

Mon Mar  7 17:42:32 UTC 2022

localhost:~/ex2# nano a1
localhost:~/ex2# cat a1
     March 2022

Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31


Mon Mar  7 17:42:32 UTC 2022
```

```
localhost:~/ex2# cat -n a1
     1       March 2022
     2
     3  Su Mo Tu We Th Fr Sa
     4         1  2  3  4  5
     5   6  7  8  9 10 11 12
     6  13 14 15 16 17 18 19
     7  20 21 22 23 24 25 26
     8  27 28 29 30 31
     9
    10
    11  Mon Mar  7 17:42:32 UTC 2022
```

```
#2-7 Use "nl -w10" to update the width. (default is 6)
localhost:~/ex2# nl -w6 a1
     1       March 2022

     2  Su Mo Tu We Th Fr Sa
     3         1  2  3  4  5
     4   6  7  8  9 10 11 12
     5  13 14 15 16 17 18 19
     6  20 21 22 23 24 25 26
     7  27 28 29 30 31
     8

     9  Mon Mar  7 17:42:32 UTC 2022
localhost:~/ex2# nl -w10 a1
         1       March 2022

         2      Su Mo Tu We Th Fr Sa
         3             1  2  3  4  5
         4       6  7  8  9 10 11 12
         5      13 14 15 16 17 18 19
         6      20 21 22 23 24 25 26
         7      27 28 29 30 31
         8

         9      Mon Mar  7 17:42:32 UTC 2022
localhost:~/ex2# nl -w6 a1
     1       March 2022

     2  Su Mo Tu We Th Fr Sa
     3         1  2  3  4  5
     4   6  7  8  9 10 11 12
     5  13 14 15 16 17 18 19
     6  20 21 22 23 24 25 26
     7  27 28 29 30 31
     8

     9  Mon Mar  7 17:42:32 UTC 2022
```

```
localhost:~/ex2# nl -w10 a1
         1	     March 2022

         2	Su Mo Tu We Th Fr Sa
         3	       1  2  3  4  5
         4	 6  7  8  9 10 11 12
         5	13 14 15 16 17 18 19
         6	20 21 22 23 24 25 26
         7	27 28 29 30 31
         8

         9	Mon Mar  7 17:42:32 UTC 2022
localhost:~/ex2# nl -w6 a1
     1	     March 2022

     2	Su Mo Tu We Th Fr Sa
     3	       1  2  3  4  5
     4	 6  7  8  9 10 11 12
     5	13 14 15 16 17 18 19
     6	20 21 22 23 24 25 26
     7	27 28 29 30 31
     8

     9	Mon Mar  7 17:42:32 UTC 2022
localhost:~/ex2# nl -w10 a1
         1	     March 2022

         2	Su Mo Tu We Th Fr Sa
         3	       1  2  3  4  5
         4	 6  7  8  9 10 11 12
         5	13 14 15 16 17 18 19
         6	20 21 22 23 24 25 26
         7	27 28 29 30 31
         8

         9	Mon Mar  7 17:42:32 UTC 2022

#2-8 Use -i option for stepping (increment)
bchoi@pop-os:~/d3$ nl a1
     1	    January 2022

     2	Su Mo Tu We Th Fr Sa
     3	                   1
     4	 2  3  4  5  6  7  8
     5	 9 10 11 12 13 14 15
     6	16 17 18 19 20 21 22
     7	23 24 25 26 27 28 29
     8	30 31

     9	bchoi    :1           2022-01-29 10:56 (:1)
bchoi@pop-os:~/d3$ nl -i2 a1
```

```
     1        January 2022

     3      Su Mo Tu We Th Fr Sa
     5                       1
     7       2  3  4  5  6  7  8
     9       9 10 11 12 13 14 15
    11      16 17 18 19 20 21 22
    13      23 24 25 26 27 28 29
    15      30 31

    17      bchoi    :1           2022-01-29 10:56 (:1)
bchoi@pop-os:~/d3$ nl -i3 a1
     1        January 2022

     4      Su Mo Tu We Th Fr Sa
     7                       1
    10       2  3  4  5  6  7  8
    13       9 10 11 12 13 14 15
    16      16 17 18 19 20 21 22
    19      23 24 25 26 27 28 29
    22      30 31

    25      bchoi    :1           2022-01-29 10:56 (:1)
bchoi@pop-os:~/d3$ nl -i5 a1
     1        January 2022

     6      Su Mo Tu We Th Fr Sa
    11                       1
    16       2  3  4  5  6  7  8
    21       9 10 11 12 13 14 15
    26      16 17 18 19 20 21 22
    31      23 24 25 26 27 28 29
    36      30 31

    41      bchoi    :1           2022-01-29 10:56 (:1)
```

```
#2-9 Specify where to begin

# Increase the line number by 10.

localhost:~/ex2# nl -i10 a1
     1      March 2022

    11  Su Mo Tu We Th Fr Sa
    21        1  2  3  4  5
    31   6  7  8  9 10 11 12
    41  13 14 15 16 17 18 19
    51  20 21 22 23 24 25 26
    61  27 28 29 30 31
    71
    81  Mon Mar  7 17:42:32 UTC 2022

# Specify the beginning of the number with '-v'.
localhost:~/ex2# nl -i10 -v10 a1
    10       March 2022

    20  Su Mo Tu We Th Fr Sa
    30         1  2  3  4  5
    40   6  7  8  9 10 11 12
    50  13 14 15 16 17 18 19
    60  20 21 22 23 24 25 26
    70  27 28 29 30 31
    80

    90  Mon Mar  7 17:42:32 UTC 2022


localhost:~/ex2# nl -i2 -v0 a1
     0       March 2022

     2  Su Mo Tu We Th Fr Sa
     4         1  2  3  4  5
     6   6  7  8  9 10 11 12
     8  13 14 15 16 17 18 19
    10  20 21 22 23 24 25 26
    12  27 28 29 30 31
    14

    16  Mon Mar  7 17:42:32 UTC 2022

localhost:~/ex2# nl -i2 -v1 a1
     1       March 2022

     3  Su Mo Tu We Th Fr Sa
     5         1  2  3  4  5
     7   6  7  8  9 10 11 12
     9  13 14 15 16 17 18 19
    11  20 21 22 23 24 25 26
    13  27 28 29 30 31
    15

    17  Mon Mar  7 17:42:32 UTC 2022

# Save to a file.
localhost:~/ex2# nl -i2 -v1 a1 > a2
localhost:~/ex2# cat a2
     1       March 2022

     3  Su Mo Tu We Th Fr Sa
     5         1  2  3  4  5
```

```
   7   6  7  8  9 10 11 12
   9  13 14 15 16 17 18 19
  11  20 21 22 23 24 25 26
  13  27 28 29 30 31
  15

  17  Mon Mar  7 17:42:32 UTC 2022
```

#2-10 Create a7, then use nl with options -v and -i and create a10. Check and compare the two files.

```
localhost:~/ex2# cal > a7
localhost:~/ex2# nl a7 -v1 -i1 > a8
localhost:~/ex2# cat a7
     March 2022
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

localhost:~/ex2# cat a8
     1       March 2022
     2  Su Mo Tu We Th Fr Sa
     3         1  2  3  4  5
     4   6  7  8  9 10 11 12
     5  13 14 15 16 17 18 19
     6  20 21 22 23 24 25 26
     7  27 28 29 30 31
     8
```

#2-11 using --help or man

```
localhost:~/ex2# nl --help
BusyBox v1.31.1 () multi-call binary.

Usage: nl [OPTIONS] [FILE]...

Write FILEs to standard output with line numbers added

        -b STYLE        Which lines to number - a: all, t: nonempty, n: none
        -i N            Line number increment
        -s STRING       Use STRING as a line number separator
        -v N            Start from N
        -w N            Width of line numbers

localhost:~/ex2# help nl
Built-in commands:
------------------
        . : [ [[ alias bg break cd chdir command continue echo eval exec
        exit export false fg getopts hash help history jobs kill let
        local printf pwd read readonly return set shift source test times
```

trap true type ulimit umask unalias unset wait

==

Ex2 Summary

tee -a
nl = cat -b
-i10 -v100 -1000 -w10

To get help, use 'tee --help' and 'nl --help'

==

Exercise 3: head & tail

time: 15 minutes
==

#3-1 Make a new directory ex3 and change the working directory to ex3.

```
localhost:~# mkdir ex3 && cd ex3
localhost:~/ex3#
```

#3-2 Create two files called c1. Use this file to create another file called c2.

```
localhost:~/ex3# cat > c1
Sydney
Melbourne
Brisbane
Perth
Adelaide
localhost:~/ex3# cat c1
Sydney
Melbourne
Brisbane
Perth
Adelaide

localhost:~/ex3# cat c1 > c2
localhost:~/ex3# cat c1 >> c2
localhost:~/ex3# cat c1 >> c2
localhost:~/ex3# cat c1 >> c2
localhost:~/ex3# nl c2
     1  Sydney
     2  Melbourne
     3  Brisbane
     4  Perth
     5  Adelaide
     6  Sydney
     7  Melbourne
     8  Brisbane
     9  Perth
    10  Adelaide
    11  Sydney
    12  Melbourne
    13  Brisbane
    14  Perth
    15  Adelaide
    16  Sydney
    17  Melbourne
    18  Brisbane
    19  Perth
    20  Adelaide
```

#3-3 Overwrite c1 with c2

```
localhost:~/ex3# nl c2 > c1
localhost:~/ex3# nl c1
     1       1  Sydney
     2       2  Melbourne
     3       3  Brisbane
     4       4  Perth
     5       5  Adelaide
     6       6  Sydney
     7       7  Melbourne
     8       8  Brisbane
     9       9  Perth
    10      10  Adelaide
    11      11  Sydney
    12      12  Melbourne
    13      13  Brisbane
    14      14  Perth
    15      15  Adelaide
    16      16  Sydney
    17      17  Melbourne
    18      18  Brisbane
    19      19  Perth
    20      20  Adelaide
```

#3-4 Using 'seq' command, create a file named nums1 and modify it to include the line reference.

```
# Sequence starts with 100, increment by 100, and up to 1000
ocalhost:~/ex3# seq 100 100 1000
100
200
300
400
500
600
700
800
900
1000

# Save the above sequence to a file named nums1

localhost:~/ex3# seq 100 100 1000 > nl > nums1
localhost:~/ex3# cat nums1
100
200
300
400
500
600
700
800
```

```
900
1000
```

# Include the line reference in the saved file by using 'nl' command with the nums1 file.

```
localhost:~/ex3# nl nums1 > nums2
localhost:~/ex3# cat nums2
     1  100
     2  200
     3  300
     4  400
     5  500
     6  600
     7  700
     8  800
     9  900
    10  1000
```

#3-5 Use the 'head' command to check the first ten lines (default is head -n 10). The default command will give the same result as '-n 10'.

```
ocalhost:~/ex3# head nums2
     1  100
     2  200
     3  300
     4  400
     5  500
     6  600
     7  700
     8  800
     9  900
    10  1000
localhost:~/ex3# head -n 10 nums2
     1  100
     2  200
     3  300
     4  400
     5  500
     6  600
     7  700
     8  800
     9  900
    10  1000
```

#3-6 Specify the number of lines you want to view.

```
localhost:~/ex3# head -n 3 nums2
     1  100
     2  200
     3  300
localhost:~/ex3# head -n 5 nums2
```

```
     1  100
     2  200
     3  300
     4  400
     5  500
localhost:~/ex3# head -n 7 nums2
     1  100
     2  200
     3  300
     4  400
     5  500
     6  600
     7  700
```

#3-7 Avoid using minus as this can be confusing. Using a minus number will drop the specified number of lines at the end.

```
ocalhost:~/ex3# head -n -2 nums2
     1  100
     2  200
     3  300
     4  400
     5  500
     6  600
     7  700
     8  800
localhost:~/ex3# head -n -8 nums2
     1  100
     2  200
```

#3-8 Use "head -n 5" on two files to get the first 5 lines

```
localhost:~/ex3# cat c1
     1  Sydney
     2  Melbourne
     3  Brisbane
     4  Perth
     5  Adelaide
     6  Sydney
     7  Melbourne
     8  Brisbane
     9  Perth
    10  Adelaide
    11  Sydney
    12  Melbourne
    13  Brisbane
    14  Perth
    15  Adelaide
    16  Sydney
    17  Melbourne
    18  Brisbane
    19  Perth
    20  Adelaide
localhost:~/ex3# cat nums2
     1  100
     2  200
     3  300
     4  400
     5  500
     6  600
     7  700
     8  800
     9  900
    10  1000
localhost:~/ex3# head -n 4 c1 nums2
==> c1 <==
     1  Sydney
     2  Melbourne
     3  Brisbane
     4  Perth

==> nums2 <==
     1  100
     2  200
     3  300
     4  400
```

#3-9 Drop the last 9 lines and display the result.

```
localhost:~/ex3# head -n -9 c1 nums2
==> c1 <==
     1  Sydney
     2  Melbourne
     3  Brisbane
     4  Perth
     5  Adelaide
     6  Sydney
     7  Melbourne
     8  Brisbane
     9  Perth
    10  Adelaide
    11  Sydney

==> nums2 <==
     1  100
```

#3-10 Use 'head -n -q' to remove the header. To remove the header, use the '-q' option

```
localhost:~/ex3# head -n 5 -q c1 nums2
     1  Sydney
     2  Melbourne
     3  Brisbane
     4  Perth
     5  Adelaide
     1  100
     2  200
     3  300
     4  400
     5  500
```

#3-11 Save the above result to a file using '>'.

```
localhost:~/ex3# head -n 5 -q c1 nums2 > cn1
localhost:~/ex3# cat cn1
     1  Sydney
     2  Melbourne
     3  Brisbane
     4  Perth
     5  Adelaide
     1  100
     2  200
     3  300
     4  400
     5  500
```

#3-12 Tail prints the output of the last 10 lines

```
ocalhost:~/ex3# tail c1
    11  Sydney
    12  Melbourne
    13  Brisbane
    14  Perth
    15  Adelaide
    16  Sydney
    17  Melbourne
    18  Brisbane
    19  Perth
    20  Adelaide

localhost:~/ex3# tail cn1
     1  Sydney
     2  Melbourne
     3  Brisbane
     4  Perth
     5  Adelaide
     1  100
     2  200
     3  300
     4  400
     5  500
```

#3-13 Tail has minus default. Check this out.

```
localhost:~/ex3# tail -n 5 cn1
     1  100
     2  200
     3  300
     4  400
     5  500
localhost:~/ex3# tail -n -5 cn1
     1  100
     2  200
     3  300
     4  400
     5  500
```

# If +11 is used, that means from line 11 to the end of the line.

```
localhost:~/ex3# tail -n +11 c1
    11  Sydney
    12  Melbourne
    13  Brisbane
    14  Perth
    15  Adelaide
    16  Sydney
    17  Melbourne
    18  Brisbane
    19  Perth
    20  Adelaide
```

#3-14 Practice with c1

```
localhost:~/ex3# tail c1
    11  Sydney
    12  Melbourne
    13  Brisbane
    14  Perth
    15  Adelaide
    16  Sydney
    17  Melbourne
    18  Brisbane
    19  Perth
    20  Adelaide
localhost:~/ex3# tail -n 5 c1
    16  Sydney
    17  Melbourne
    18  Brisbane
    19  Perth
    20  Adelaide
localhost:~/ex3# tail -n +5 c1
     5  Adelaide
     6  Sydney
     7  Melbourne
     8  Brisbane
     9  Perth
    10  Adelaide
    11  Sydney
    12  Melbourne
    13  Brisbane
    14  Perth
    15  Adelaide
    16  Sydney
    17  Melbourne
    18  Brisbane
    19  Perth
    20  Adelaide
```

#3-15 Run single command on two files.

```
localhost:~/ex3# tail -n 5 c1 nums2
==> c1 <==
    16  Sydney
    17  Melbourne
    18  Brisbane
    19  Perth
    20  Adelaide

==> nums2 <==
     6  600
     7  700
     8  800
     9  900
    10  1000

localhost:~/ex3# tail -n 5 -q c1 nums2
    16  Sydney
    17  Melbourne
    18  Brisbane
    19  Perth
    20  Adelaide
     6  600
     7  700
     8  800
     9  900
    10  1000
```

#3-16 Use the '-c' option to print in bytes. '-c' option is not used a lot, but it is good to know this option exists.

```
localhost:~/ex3# tail -c 10 nums2
  10    1000
localhost:~/ex3# tail -c 20 nums2
  9     900
    10  1000
```

```
localhost:~/ex3# tail -c 30 nums2
 8      800
    9  900
   10  1000
localhost:~/ex3# tail -c 50 nums2
      600
    7  700
    8  800
    9  900
   10  1000
```

#3-17 For more help, refer to tail/head --help for KB/MB output

```
localhost:~/ex3# tail --help
```

==

Optional - If you are using your own Linux, you can perform the following task.

#3-18 Tail has a tracking function. A special function of the tail.

On the first console, use the 'tail -f' option to track the changes.

```
bchoi@pop-os:~/d3$ tail -f numbers
     6      600
     7      700
     8      800
     9      900
    10      1000
    11      1100
    12      1200
    13      1300
    14      1400
    15      1500
```

Open a second command console and append two lines using echo.

```
bchoi@pop-os:~/d3$ echo "G'day" >> numbers
bchoi@pop-os:~/d3$ echo "Mate" >> numbers
```

On the first command console, you will see the information getting added to the last lines.

```
bchoi@pop-os:~/d3$ tail -f numbers
     6      600
     7      700
     8      800
     9      900
    10      1000
    11      1100
    12      1200
    13      1300
    14      1400
    15      1500
G'day
Mate

==================================================================================
```

Ex3 Summary

```
head file //print first 10 lines
head -n 10 file //print first 10 lines
head -10 file //print first 10 lines (works, but not recommended)
head -n +10 file //print first 10 lines
head -n -5 file //Remove the last 5 lines

tail file //print the last 10 lines
tail -n 10 file //print the last 10 lines
tail -10 file //print the last 10 lines
tail -n -10 file //print from the last 10th line
tail -n +5 file //print from the 5th line to the last item

tail -f //monitor the real-time file change
==================================================================================
```

Exercise 4: split

time: 12 minutes

```
================================================================================
```

#4-1a Make and move to the new directory. Copy /etc/services file. and check the size of the file.

```
[jdoe@f33sn1 ~]$ mkdir d4 && cd d4
[jdoe@f33sn1 d4]$ cp /etc/services m

-----------------------------------------------------
```

#4-1b If using a website, you have to create the file you
Create a file size that is between 500k~999k

```
localhost:~/d7# cp /etc/services k
localhost:~/d7# nano multiple.sh

#!/bin/bash
for i in {1..10}; do cat k k >> m; done

localhost:~/d7# bash multiple.sh
localhost:~/d7# ls -lh
total 304K
-rw-r--r--    1 root     root       14.1K Feb  7 15:36 k
-rw-r--r--    1 root     root      282.5K Feb  7 17:25 m
-rw-r--r--    1 root     root          52 Feb  7 17:20 multiple.sh
localhost:~/d7# ls -ll
total 304
-rw-r--r--    1 root     root         14464 Feb  7 15:36 k
-rw-r--r--    1 root     root        289280 Feb  7 17:25 m
-rw-r--r--    1 root     root            52 Feb  7 17:20 multiple.sh
-----------------------------------------------------
```

#4-2 make lh alias

```
[jdoe@f33sn1 d4]$ alias lh='ls -Alh'
[jdoe@f33sn1 d4]$ lh
total 680K
-rw-r--r--. 1 jdoe jdoe 677K Feb  7 19:34 m
```

#4-3 Check the lines of file m

```
[jdoe@f33sn1 d4]$ nl m
[jdoe@f33sn1 d4]$ wc -l m
11473 m
[jdoe@f33sn1 d4]$ lh m
-rw-r--r--. 1 jdoe jdoe 677K Feb  7 19:34 m
[jdoe@f33sn1 d4]$ ll m
-rw-r--r--. 1 jdoe jdoe 692252 Feb  7 19:34 m
```

#4-4 Split the file m

```
[jdoe@f33sn1 d4]$ split m
[jdoe@f33sn1 d4]$ lh
total 1.4M
```

```
-rw-r--r--. 1 jdoe jdoe 677K Feb  7 19:34 m
-rw-r--r--. 1 jdoe jdoe  56K Feb  7 19:38 xaa
-rw-r--r--. 1 jdoe jdoe  57K Feb  7 19:38 xab
-rw-r--r--. 1 jdoe jdoe  57K Feb  7 19:38 xac
-rw-r--r--. 1 jdoe jdoe  57K Feb  7 19:38 xad
-rw-r--r--. 1 jdoe jdoe  55K Feb  7 19:38 xae
-rw-r--r--. 1 jdoe jdoe  57K Feb  7 19:38 xaf
-rw-r--r--. 1 jdoe jdoe  59K Feb  7 19:38 xag
-rw-r--r--. 1 jdoe jdoe  63K Feb  7 19:38 xah
-rw-r--r--. 1 jdoe jdoe  63K Feb  7 19:38 xai
-rw-r--r--. 1 jdoe jdoe  64K Feb  7 19:38 xaj
-rw-r--r--. 1 jdoe jdoe  64K Feb  7 19:38 xak
-rw-r--r--. 1 jdoe jdoe  30K Feb  7 19:38 xal
```

#4-5 Chek the total lines in m and all files combined in xa*, should be the same number.

```
[jdoe@f33sn1 d4]$ wc -l m
11473 m

[jdoe@f33sn1 d4]$ wc -l xa*
  1000 xaa
  1000 xab
  1000 xac
  1000 xad
  1000 xae
  1000 xaf
  1000 xag
  1000 xah
  1000 xai
  1000 xaj
  1000 xak
   473 xal
 11473 total
```

#4-6 Merge all xa* files into m2 using cat command

```
[jdoe@f33sn1 d4]$ cat xa* > m2
[jdoe@f33sn1 d4]$ ls -lh m*
-rw-r--r--. 1 jdoe jdoe 677K Feb  7 19:34 m
-rw-r--r--. 1 jdoe jdoe 677K Feb  7 19:40 m2
```

#4-7 Use the "diff" command to check the files m vs m2. If the files are the same, no output should be observed.

```
[jdoe@f33sn1 d4]$ diff m m2
```

Quick diff check

```
localhost:~/d7# cat a
a
b
c
```

```
localhost:~/d7# cat b
a
x
z
localhost:~/d7# diff a b
--- a
+++ b
@@ -1,3 +1,3 @@
 a
-b
-c
+x
+z
------------------------
```

#4-8 Remove all splited files starting with xa*

```
[jdoe@f33sn1 d4]$ lh
total 2.1M
-rw-r--r--. 1 jdoe jdoe 677K Feb  7 19:34 m
-rw-r--r--. 1 jdoe jdoe 677K Feb  7 19:40 m2
-rw-r--r--. 1 jdoe jdoe  56K Feb  7 19:38 xaa
-rw-r--r--. 1 jdoe jdoe  57K Feb  7 19:38 xab
-rw-r--r--. 1 jdoe jdoe  57K Feb  7 19:38 xac
-rw-r--r--. 1 jdoe jdoe  57K Feb  7 19:38 xad
-rw-r--r--. 1 jdoe jdoe  55K Feb  7 19:38 xae
-rw-r--r--. 1 jdoe jdoe  57K Feb  7 19:38 xaf
-rw-r--r--. 1 jdoe jdoe  59K Feb  7 19:38 xag
-rw-r--r--. 1 jdoe jdoe  63K Feb  7 19:38 xah
-rw-r--r--. 1 jdoe jdoe  63K Feb  7 19:38 xai
-rw-r--r--. 1 jdoe jdoe  64K Feb  7 19:38 xaj
-rw-r--r--. 1 jdoe jdoe  64K Feb  7 19:38 xak
-rw-r--r--. 1 jdoe jdoe  30K Feb  7 19:38 xal
[jdoe@f33sn1 d4]$ rm xa*
[jdoe@f33sn1 d4]$ lh
total 1.4M
-rw-r--r--. 1 jdoe jdoe 677K Feb  7 19:34 m
-rw-r--r--. 1 jdoe jdoe 677K Feb  7 19:40 m2
```

#4-9 Use decimal option (-d) to split the file and use names starting with x.

```
[jdoe@f33sn1 d4]$ split -d m
[jdoe@f33sn1 d4]$ lh
total 2.1M
-rw-r--r--. 1 jdoe jdoe 677K Feb  7 19:34 m
-rw-r--r--. 1 jdoe jdoe 677K Feb  7 19:40 m2
-rw-r--r--. 1 jdoe jdoe  56K Feb  7 19:43 x00
-rw-r--r--. 1 jdoe jdoe  57K Feb  7 19:43 x01
-rw-r--r--. 1 jdoe jdoe  57K Feb  7 19:43 x02
-rw-r--r--. 1 jdoe jdoe  57K Feb  7 19:43 x03
-rw-r--r--. 1 jdoe jdoe  55K Feb  7 19:43 x04
-rw-r--r--. 1 jdoe jdoe  57K Feb  7 19:43 x05
-rw-r--r--. 1 jdoe jdoe  59K Feb  7 19:43 x06
```

```
-rw-r--r--. 1 jdoe jdoe  63K Feb  7 19:43 x07
-rw-r--r--. 1 jdoe jdoe  63K Feb  7 19:43 x08
-rw-r--r--. 1 jdoe jdoe  64K Feb  7 19:43 x09
-rw-r--r--. 1 jdoe jdoe  64K Feb  7 19:43 x10
-rw-r--r--. 1 jdoe jdoe  30K Feb  7 19:43 x11

[jdoe@f33sn1 d4]$ ll
total 2060
-rw-r--r--. 1 jdoe jdoe 692252 Feb  7 19:34 m
-rw-r--r--. 1 jdoe jdoe 692252 Feb  7 19:40 m2
-rw-r--r--. 1 jdoe jdoe  56600 Feb  7 19:43 x00
-rw-r--r--. 1 jdoe jdoe  58335 Feb  7 19:43 x01
-rw-r--r--. 1 jdoe jdoe  57905 Feb  7 19:43 x02
-rw-r--r--. 1 jdoe jdoe  57807 Feb  7 19:43 x03
-rw-r--r--. 1 jdoe jdoe  55935 Feb  7 19:43 x04
-rw-r--r--. 1 jdoe jdoe  57422 Feb  7 19:43 x05
-rw-r--r--. 1 jdoe jdoe  60089 Feb  7 19:43 x06
-rw-r--r--. 1 jdoe jdoe  63727 Feb  7 19:43 x07
-rw-r--r--. 1 jdoe jdoe  64022 Feb  7 19:43 x08
-rw-r--r--. 1 jdoe jdoe  64561 Feb  7 19:43 x09
-rw-r--r--. 1 jdoe jdoe  65159 Feb  7 19:43 x10
-rw-r--r--. 1 jdoe jdoe  30690 Feb  7 19:43 x11

#4-10 Use -a option to specify the length of the file name.
[jdoe@f33sn1 d4]$ split -d -a 5 m
[jdoe@f33sn1 d4]$ lh
total 2.7M
-rw-r--r--. 1 jdoe jdoe 677K Feb  7 19:34 m
-rw-r--r--. 1 jdoe jdoe 677K Feb  7 19:40 m2
-rw-r--r--. 1 jdoe jdoe  56K Feb  7 19:43 x00
-rw-r--r--. 1 jdoe jdoe  56K Feb  7 19:44 x00000
-rw-r--r--. 1 jdoe jdoe  57K Feb  7 19:44 x00001
-rw-r--r--. 1 jdoe jdoe  57K Feb  7 19:44 x00002
-rw-r--r--. 1 jdoe jdoe  57K Feb  7 19:44 x00003
-rw-r--r--. 1 jdoe jdoe  55K Feb  7 19:44 x00004
-rw-r--r--. 1 jdoe jdoe  57K Feb  7 19:44 x00005
-rw-r--r--. 1 jdoe jdoe  59K Feb  7 19:44 x00006
-rw-r--r--. 1 jdoe jdoe  63K Feb  7 19:44 x00007
-rw-r--r--. 1 jdoe jdoe  63K Feb  7 19:44 x00008
-rw-r--r--. 1 jdoe jdoe  64K Feb  7 19:44 x00009
-rw-r--r--. 1 jdoe jdoe  64K Feb  7 19:44 x00010
-rw-r--r--. 1 jdoe jdoe  30K Feb  7 19:44 x00011
-rw-r--r--. 1 jdoe jdoe  57K Feb  7 19:43 x01
-rw-r--r--. 1 jdoe jdoe  57K Feb  7 19:43 x02
-rw-r--r--. 1 jdoe jdoe  57K Feb  7 19:43 x03
-rw-r--r--. 1 jdoe jdoe  55K Feb  7 19:43 x04
-rw-r--r--. 1 jdoe jdoe  57K Feb  7 19:43 x05
-rw-r--r--. 1 jdoe jdoe  59K Feb  7 19:43 x06
-rw-r--r--. 1 jdoe jdoe  63K Feb  7 19:43 x07
-rw-r--r--. 1 jdoe jdoe  63K Feb  7 19:43 x08
-rw-r--r--. 1 jdoe jdoe  64K Feb  7 19:43 x09
```

```
-rw-r--r--. 1 jdoe jdoe  64K Feb  7 19:43 x10
-rw-r--r--. 1 jdoe jdoe  30K Feb  7 19:43 x11
```

#4-11 Remove files and only leave the original file, m.

```
[jdoe@f33sn1 d4]$ rm -f x*
[jdoe@f33sn1 d4]$ lh
total 1.4M
-rw-r--r--. 1 jdoe jdoe 677K Feb  7 19:34 m
-rw-r--r--. 1 jdoe jdoe 677K Feb  7 19:40 m2
[jdoe@f33sn1 d4]$ rm -f m2
[jdoe@f33sn1 d4]$ lh
total 680K
-rw-r--r--. 1 jdoe jdoe 677K Feb  7 19:34 m
```

#4-12 add the file extension names to the file.

```
[jdoe@f33sn1 d4]$ split -d --additional-suffix=.txt m
[jdoe@f33sn1 d4]$ lh
total 1.4M
-rw-r--r--. 1 jdoe jdoe 677K Feb  7 19:34 m
-rw-r--r--. 1 jdoe jdoe  56K Feb  7 19:48 x00.txt
-rw-r--r--. 1 jdoe jdoe  57K Feb  7 19:48 x01.txt
-rw-r--r--. 1 jdoe jdoe  57K Feb  7 19:48 x02.txt
-rw-r--r--. 1 jdoe jdoe  57K Feb  7 19:48 x03.txt
-rw-r--r--. 1 jdoe jdoe  55K Feb  7 19:48 x04.txt
-rw-r--r--. 1 jdoe jdoe  57K Feb  7 19:48 x05.txt
-rw-r--r--. 1 jdoe jdoe  59K Feb  7 19:48 x06.txt
-rw-r--r--. 1 jdoe jdoe  63K Feb  7 19:48 x07.txt
-rw-r--r--. 1 jdoe jdoe  63K Feb  7 19:48 x08.txt
-rw-r--r--. 1 jdoe jdoe  64K Feb  7 19:48 x09.txt
-rw-r--r--. 1 jdoe jdoe  64K Feb  7 19:48 x10.txt
-rw-r--r--. 1 jdoe jdoe  30K Feb  7 19:48 x11.txt

[jdoe@f33sn1 d4]$ rm -rf x*
[jdoe@f33sn1 d4]$ lh
total 680K
-rw-r--r--. 1 jdoe jdoe 677K Feb  7 19:34 m
```

#4-13 Use --additional-suffix option and file name prefix to give a more meaningful name.

```
[jdoe@f33sn1 d4]$ split -d --additional-suffix=.txt m backup_
[jdoe@f33sn1 d4]$ lh
total 1.4M
-rw-r--r--. 1 jdoe jdoe  56K Feb  7 19:49 backup_00.txt
-rw-r--r--. 1 jdoe jdoe  57K Feb  7 19:49 backup_01.txt
-rw-r--r--. 1 jdoe jdoe  57K Feb  7 19:49 backup_02.txt
-rw-r--r--. 1 jdoe jdoe  57K Feb  7 19:49 backup_03.txt
-rw-r--r--. 1 jdoe jdoe  55K Feb  7 19:49 backup_04.txt
-rw-r--r--. 1 jdoe jdoe  57K Feb  7 19:49 backup_05.txt
-rw-r--r--. 1 jdoe jdoe  59K Feb  7 19:49 backup_06.txt
-rw-r--r--. 1 jdoe jdoe  63K Feb  7 19:49 backup_07.txt
-rw-r--r--. 1 jdoe jdoe  63K Feb  7 19:49 backup_08.txt
-rw-r--r--. 1 jdoe jdoe  64K Feb  7 19:49 backup_09.txt
```

```
-rw-r--r--. 1 jdoe jdoe  64K Feb  7 19:49 backup_10.txt
-rw-r--r--. 1 jdoe jdoe  30K Feb  7 19:49 backup_11.txt
-rw-r--r--. 1 jdoe jdoe 677K Feb  7 19:34 m
```

#4-14 Check the splited files are divided by 1000 lines each.

```
[jdoe@f33sn1 d4]$ wc -l *
   1000 backup_00.txt
   1000 backup_01.txt
   1000 backup_02.txt
   1000 backup_03.txt
   1000 backup_04.txt
   1000 backup_05.txt
   1000 backup_06.txt
   1000 backup_07.txt
   1000 backup_08.txt
   1000 backup_09.txt
   1000 backup_10.txt
    473 backup_11.txt
  11473 m
  22946 total

[jdoe@f33sn1 d4]$ lh
total 680K
-rw-r--r--. 1 jdoe jdoe 677K Feb  7 19:34 m
```

#4-15 Use -l option to change the number of lines to save in each file.

```
[jdoe@f33sn1 d4]$ split -d --additional-suffix=.txt -l 3000 m backup_
[jdoe@f33sn1 d4]$ ll
total 1364
-rw-r--r--. 1 jdoe jdoe 172840 Feb  7 19:53 backup_00.txt
-rw-r--r--. 1 jdoe jdoe 171164 Feb  7 19:53 backup_01.txt
-rw-r--r--. 1 jdoe jdoe 187838 Feb  7 19:53 backup_02.txt
-rw-r--r--. 1 jdoe jdoe 160410 Feb  7 19:53 backup_03.txt
-rw-r--r--. 1 jdoe jdoe 692252 Feb  7 19:34 m

[jdoe@f33sn1 d4]$ wc -l *
   3000 backup_00.txt
   3000 backup_01.txt
   3000 backup_02.txt
   2473 backup_03.txt
  11473 m
  22946 total

[jdoe@f33sn1 d4]$ rm -rf b*
[jdoe@f33sn1 d4]$ lh
total 680K
-rw-r--r--. 1 jdoe jdoe 677K Feb  7 19:34 m
```

#4-16 Split a large file using bytes, option -C.

```
[jdoe@f33sn1 d4]$ split -d --additional-suffix=.txt -C 30000 m backup_
```

```
[jdoe@f33sn1 d4]$ lh
total 1.4M
-rw-r--r--. 1 jdoe jdoe  30K Feb  7 19:57 backup_00.txt
-rw-r--r--. 1 jdoe jdoe  30K Feb  7 19:57 backup_01.txt
-rw-r--r--. 1 jdoe jdoe  30K Feb  7 19:57 backup_02.txt
-rw-r--r--. 1 jdoe jdoe  30K Feb  7 19:57 backup_03.txt
-rw-r--r--. 1 jdoe jdoe  30K Feb  7 19:57 backup_04.txt
-rw-r--r--. 1 jdoe jdoe  30K Feb  7 19:57 backup_05.txt
-rw-r--r--. 1 jdoe jdoe  30K Feb  7 19:57 backup_06.txt
-rw-r--r--. 1 jdoe jdoe  30K Feb  7 19:57 backup_07.txt
-rw-r--r--. 1 jdoe jdoe  30K Feb  7 19:57 backup_08.txt
-rw-r--r--. 1 jdoe jdoe  30K Feb  7 19:57 backup_09.txt
-rw-r--r--. 1 jdoe jdoe  30K Feb  7 19:57 backup_10.txt
-rw-r--r--. 1 jdoe jdoe  30K Feb  7 19:57 backup_11.txt
-rw-r--r--. 1 jdoe jdoe  30K Feb  7 19:57 backup_12.txt
-rw-r--r--. 1 jdoe jdoe  30K Feb  7 19:57 backup_13.txt
-rw-r--r--. 1 jdoe jdoe  30K Feb  7 19:57 backup_14.txt
-rw-r--r--. 1 jdoe jdoe  30K Feb  7 19:57 backup_15.txt
-rw-r--r--. 1 jdoe jdoe  30K Feb  7 19:57 backup_16.txt
-rw-r--r--. 1 jdoe jdoe  30K Feb  7 19:57 backup_17.txt
-rw-r--r--. 1 jdoe jdoe  30K Feb  7 19:57 backup_18.txt
-rw-r--r--. 1 jdoe jdoe  30K Feb  7 19:57 backup_19.txt
-rw-r--r--. 1 jdoe jdoe  30K Feb  7 19:57 backup_20.txt
-rw-r--r--. 1 jdoe jdoe  30K Feb  7 19:57 backup_21.txt
-rw-r--r--. 1 jdoe jdoe  30K Feb  7 19:57 backup_22.txt
-rw-r--r--. 1 jdoe jdoe 3.1K Feb  7 19:57 backup_23.txt
-rw-r--r--. 1 jdoe jdoe 677K Feb  7 19:34 m
[jdoe@f33sn1 d4]$ rm -rf b*
[jdoe@f33sn1 d4]$ ls
m
```

#4-17 Specify the number of files and split the file into desired number of files, using option -n.

```
[jdoe@f33sn1 d4]$ split -d --additional-suffix=.txt -n 3 m backup_
[jdoe@f33sn1 d4]$ lh
total 1.4M
-rw-r--r--. 1 jdoe jdoe 226K Feb  7 20:00 backup_00.txt
-rw-r--r--. 1 jdoe jdoe 226K Feb  7 20:00 backup_01.txt
-rw-r--r--. 1 jdoe jdoe 226K Feb  7 20:00 backup_02.txt
-rw-r--r--. 1 jdoe jdoe 677K Feb  7 19:34 m
[jdoe@f33sn1 d4]$ rm -rf b*
[jdoe@f33sn1 d4]$ ls
m
[jdoe@f33sn1 d4]$ split -d --additional-suffix=.txt -n 10 m backup_
[jdoe@f33sn1 d4]$ lh
total 1.4M
-rw-r--r--. 1 jdoe jdoe  68K Feb  7 20:00 backup_00.txt
-rw-r--r--. 1 jdoe jdoe  68K Feb  7 20:00 backup_01.txt
-rw-r--r--. 1 jdoe jdoe  68K Feb  7 20:00 backup_02.txt
-rw-r--r--. 1 jdoe jdoe  68K Feb  7 20:00 backup_03.txt
-rw-r--r--. 1 jdoe jdoe  68K Feb  7 20:00 backup_04.txt
```

```
-rw-r--r--. 1 jdoe jdoe  68K Feb  7 20:00 backup_05.txt
-rw-r--r--. 1 jdoe jdoe  68K Feb  7 20:00 backup_06.txt
-rw-r--r--. 1 jdoe jdoe  68K Feb  7 20:00 backup_07.txt
-rw-r--r--. 1 jdoe jdoe  68K Feb  7 20:00 backup_08.txt
-rw-r--r--. 1 jdoe jdoe  68K Feb  7 20:00 backup_09.txt
-rw-r--r--. 1 jdoe jdoe 677K Feb  7 19:34 m
[jdoe@f33sn1 d4]$ rm -rf b*
[jdoe@f33sn1 d4]$ ls
m
```

#4-18 Change the numeric sequence by using the --numeric-suffixes option, this example starts the file with the number 33 and ascending.

```
[jdoe@f33sn1 d4]$ split -d --additional-suffix=.txt -n 5 --numeric-suffixes=33 m backup_
[jdoe@f33sn1 d4]$ lh
total 1.4M
-rw-r--r--. 1 jdoe jdoe 136K Feb  7 20:03 backup_33.txt
-rw-r--r--. 1 jdoe jdoe 136K Feb  7 20:03 backup_34.txt
-rw-r--r--. 1 jdoe jdoe 136K Feb  7 20:03 backup_35.txt
-rw-r--r--. 1 jdoe jdoe 136K Feb  7 20:03 backup_36.txt
-rw-r--r--. 1 jdoe jdoe 136K Feb  7 20:03 backup_37.txt
-rw-r--r--. 1 jdoe jdoe 677K Feb  7 19:34 m
[jdoe@f33sn1 d4]$ rm -rf b*
[jdoe@f33sn1 d4]$ ls
m
```

```
=======================
Quick revision
=======================
#4-19
[jdoe@f33sn1 d4]$ split -C 50000 m
[jdoe@f33sn1 d4]$ lh
total 1.4M
-rw-r--r--. 1 jdoe jdoe 677K Feb  7 19:34 m
-rw-r--r--. 1 jdoe jdoe  49K Feb  7 20:05 xaa
-rw-r--r--. 1 jdoe jdoe  49K Feb  7 20:05 xab
-rw-r--r--. 1 jdoe jdoe  49K Feb  7 20:05 xac
-rw-r--r--. 1 jdoe jdoe  49K Feb  7 20:05 xad
-rw-r--r--. 1 jdoe jdoe  49K Feb  7 20:05 xae
-rw-r--r--. 1 jdoe jdoe  49K Feb  7 20:05 xaf
-rw-r--r--. 1 jdoe jdoe  49K Feb  7 20:05 xag
-rw-r--r--. 1 jdoe jdoe  49K Feb  7 20:05 xah
-rw-r--r--. 1 jdoe jdoe  49K Feb  7 20:05 xai
-rw-r--r--. 1 jdoe jdoe  49K Feb  7 20:05 xaj
-rw-r--r--. 1 jdoe jdoe  49K Feb  7 20:05 xak
-rw-r--r--. 1 jdoe jdoe  49K Feb  7 20:05 xal
-rw-r--r--. 1 jdoe jdoe  49K Feb  7 20:05 xam
-rw-r--r--. 1 jdoe jdoe  42K Feb  7 20:05 xan
[jdoe@f33sn1 d4]$ rm -rf x*
[jdoe@f33sn1 d4]$ ls
m
[jdoe@f33sn1 d4]$ split -n 5 m
```

```
[jdoe@f33sn1 d4]$ lh
total 1.4M
-rw-r--r--. 1 jdoe jdoe 677K Feb  7 19:34 m
-rw-r--r--. 1 jdoe jdoe 136K Feb  7 20:06 xaa
-rw-r--r--. 1 jdoe jdoe 136K Feb  7 20:06 xab
-rw-r--r--. 1 jdoe jdoe 136K Feb  7 20:06 xac
-rw-r--r--. 1 jdoe jdoe 136K Feb  7 20:06 xad
-rw-r--r--. 1 jdoe jdoe 136K Feb  7 20:06 xae
[jdoe@f33sn1 d4]$ rm -rf x*
[jdoe@f33sn1 d4]$ ls
m
[jdoe@f33sn1 d4]$ split -n 5 --additional-suffix=.bak m
[jdoe@f33sn1 d4]$ lh
total 1.4M
-rw-r--r--. 1 jdoe jdoe 677K Feb  7 19:34 m
-rw-r--r--. 1 jdoe jdoe 136K Feb  7 20:08 xaa.bak
-rw-r--r--. 1 jdoe jdoe 136K Feb  7 20:08 xab.bak
-rw-r--r--. 1 jdoe jdoe 136K Feb  7 20:08 xac.bak
-rw-r--r--. 1 jdoe jdoe 136K Feb  7 20:08 xad.bak
-rw-r--r--. 1 jdoe jdoe 136K Feb  7 20:08 xae.bak
[jdoe@f33sn1 d4]$ rm -rf x*
[jdoe@f33sn1 d4]$ lh
total 680K
-rw-r--r--. 1 jdoe jdoe 677K Feb  7 19:34 m
[jdoe@f33sn1 d4]$ split -n 5 --additional-suffix=.bak -d m
[jdoe@f33sn1 d4]$ lh
total 1.4M
-rw-r--r--. 1 jdoe jdoe 677K Feb  7 19:34 m
-rw-r--r--. 1 jdoe jdoe 136K Feb  7 20:09 x00.bak
-rw-r--r--. 1 jdoe jdoe 136K Feb  7 20:09 x01.bak
-rw-r--r--. 1 jdoe jdoe 136K Feb  7 20:09 x02.bak
-rw-r--r--. 1 jdoe jdoe 136K Feb  7 20:09 x03.bak
-rw-r--r--. 1 jdoe jdoe 136K Feb  7 20:09 x04.bak
[jdoe@f33sn1 d4]$ rm -rf x*
[jdoe@f33sn1 d4]$ lh
total 680K
-rw-r--r--. 1 jdoe jdoe 677K Feb  7 19:34 m
[jdoe@f33sn1 d4]$ split -n 5 --additional-suffix=.bak --numeric-suffixes=10 -d m
[jdoe@f33sn1 d4]$ lh
total 1.4M
-rw-r--r--. 1 jdoe jdoe 677K Feb  7 19:34 m
-rw-r--r--. 1 jdoe jdoe 136K Feb  7 20:10 x10.bak
-rw-r--r--. 1 jdoe jdoe 136K Feb  7 20:10 x11.bak
-rw-r--r--. 1 jdoe jdoe 136K Feb  7 20:10 x12.bak
-rw-r--r--. 1 jdoe jdoe 136K Feb  7 20:10 x13.bak
-rw-r--r--. 1 jdoe jdoe 136K Feb  7 20:10 x14.bak
[jdoe@f33sn1 d4]$ rm -rf x*
[jdoe@f33sn1 d4]$ lh
total 680K
-rw-r--r--. 1 jdoe jdoe 677K Feb  7 19:34 m
[jdoe@f33sn1 d4]$ split -n 5 --additional-suffix=.bak --numeric-suffixes=10 -d m myservices_
```

```
[jdoe@f33sn1 d4]$ lh
total 1.4M
-rw-r--r--. 1 jdoe jdoe 677K Feb  7 20:12 m
-rw-r--r--. 1 jdoe jdoe 136K Feb  7 20:14 myservices_10.bak
-rw-r--r--. 1 jdoe jdoe 136K Feb  7 20:14 myservices_11.bak
-rw-r--r--. 1 jdoe jdoe 136K Feb  7 20:14 myservices_12.bak
-rw-r--r--. 1 jdoe jdoe 136K Feb  7 20:14 myservices_13.bak
-rw-r--r--. 1 jdoe jdoe 136K Feb  7 20:14 myservices_14.bak
[jdoe@f33sn1 d4]$ cat myservices_* > m3
[jdoe@f33sn1 d4]$ lh
total 2.0M
-rw-r--r--. 1 jdoe jdoe 677K Feb  7 20:12 m
-rw-r--r--. 1 jdoe jdoe 677K Feb  7 20:14 m3
-rw-r--r--. 1 jdoe jdoe 136K Feb  7 20:14 myservices_10.bak
-rw-r--r--. 1 jdoe jdoe 136K Feb  7 20:14 myservices_11.bak
-rw-r--r--. 1 jdoe jdoe 136K Feb  7 20:14 myservices_12.bak
-rw-r--r--. 1 jdoe jdoe 136K Feb  7 20:14 myservices_13.bak
-rw-r--r--. 1 jdoe jdoe 136K Feb  7 20:14 myservices_14.bak
[jdoe@f33sn1 d4]$ wc -l m3
11473 m3
[jdoe@f33sn1 d4]$ wc -l m
11473 m
[jdoe@f33sn1 d4]$ diff m m3
[jdoe@f33sn1 d4]$ rm -rf m*
[jdoe@f33sn1 d4]$ ls
[jdoe@f33sn1 d4]$ cp /etc/services m
[jdoe@f33sn1 d4]$ lh
total 680K
-rw-r--r--. 1 jdoe jdoe 677K Feb  7 20:15 m

======================
GETTING HELP
======================
#4-20 Use --version and --help after the command to get assistance. Get familiar with using --help.
[jdoe@f33sn1 d4]$ split --version
split (GNU coreutils) 8.32
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Written by Torbjorn Granlund and Richard M. Stallman.
[jdoe@f33sn1 d4]$ split --help
Usage: split [OPTION]... [FILE [PREFIX]]
Output pieces of FILE to PREFIXaa, PREFIXab, ...;
default size is 1000 lines, and default PREFIX is 'x'.

With no FILE, or when FILE is -, read standard input.

Mandatory arguments to long options are mandatory for short options too.
  -a, --suffix-length=N   generate suffixes of length N (default 2)
```

```
     --additional-suffix=SUFFIX  append an additional SUFFIX to file names
  -b, --bytes=SIZE        put SIZE bytes per output file
  -C, --line-bytes=SIZE   put at most SIZE bytes of records per output file
  -d                      use numeric suffixes starting at 0, not alphabetic
     --numeric-suffixes[=FROM]  same as -d, but allow setting the start value
  -x                      use hex suffixes starting at 0, not alphabetic
     --hex-suffixes[=FROM]  same as -x, but allow setting the start value
  -e, --elide-empty-files  do not generate empty output files with '-n'
     --filter=COMMAND    write to shell COMMAND; file name is $FILE
  -l, --lines=NUMBER      put NUMBER lines/records per output file
  -n, --number=CHUNKS     generate CHUNKS output files; see explanation below
  -t, --separator=SEP     use SEP instead of newline as the record separator;
                          '\0' (zero) specifies the NUL character
  -u, --unbuffered        immediately copy input to output with '-n r/...'
     --verbose           print a diagnostic just before each
                          output file is opened
     --help     display this help and exit
     --version  output version information and exit

The SIZE argument is an integer and optional unit (example: 10K is 10*1024).
Units are K,M,G,T,P,E,Z,Y (powers of 1024) or KB,MB,... (powers of 1000).
Binary prefixes can be used, too: KiB=K, MiB=M, and so on.

CHUNKS may be:
  N       split into N files based on size of input
  K/N     output Kth of N to stdout
  l/N     split into N files without splitting lines/records
  l/K/N   output Kth of N to stdout without splitting lines/records
  r/N     like 'l' but use round robin distribution
  r/K/N   likewise but only output Kth of N to stdout

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Full documentation <https://www.gnu.org/software/coreutils/split>
or available locally via: info '(coreutils) split invocation'
```

==

Ex4 Summary

```
split
-l
-C
-a
-d
--additional-suffix
--numeric-suffixes
```

==

Exercise 5: csplit & wc

time: 20 minutes

==

#5-1 Make a new directory d5, ccreate new files by copying /etc/services as s and /etc/passwd as p.

```
[jdoe@f33sn1 ~]$ mkdir d5 && cd d5
[jdoe@f33sn1 d5]$ cp /etc/services s
[jdoe@f33sn1 d5]$ cp /etc/passwd p
[jdoe@f33sn1 d5]$ wc -l s
11473 s
[jdoe@f33sn1 d5]$ wc -l p
35 p
```

----

*wc = word count

wc -c -l -w

wc --help

```
[jdoe@f33sn1 d5]$ wc s
 11473  63129 692252 s
[jdoe@f33sn1 d5]$ wc -l s
11473 s
[jdoe@f33sn1 d5]$ wc -w s
63129 s
[jdoe@f33sn1 d5]$ wc -c s
692252 s
```

#5-2 Use csplit to split the file, s.

```
[jdoe@f33sn1 d5]$  csplit s
csplit: missing operand after ‘s’
Try 'csplit --help' for more information.
[jdoe@f33sn1 d5]$ wc -l s
11473 s
[jdoe@f33sn1 d5]$ csplit s 3000
172787
519465
[jdoe@f33sn1 d5]$ wc -l xx00
2999 xx00
[jdoe@f33sn1 d5]$ ls -lh
total 1.4M
-rw-r--r--. 1 jdoe jdoe 1.8K Feb  8 11:31 p
-rw-r--r--. 1 jdoe jdoe 677K Feb  8 11:31 s
-rw-r--r--. 1 jdoe jdoe 169K Feb  8 11:37 xx00
-rw-r--r--. 1 jdoe jdoe 508K Feb  8 11:37 xx01
[jdoe@f33sn1 d5]$ alias lh='ls -Alh'
[jdoe@f33sn1 d5]$ lh x*
-rw-r--r--. 1 jdoe jdoe 169K Feb  8 11:37 xx00
-rw-r--r--. 1 jdoe jdoe 508K Feb  8 11:37 xx01

[jdoe@f33sn1 d5]$ csplit s 5000
```

```
286534
405718
[jdoe@f33sn1 d5]$ wc -l xx00
4999 xx00
[jdoe@f33sn1 d5]$ lh x*
-rw-r--r--. 1 jdoe jdoe 280K Feb  8 11:40 xx00
-rw-r--r--. 1 jdoe jdoe 397K Feb  8 11:40 xx01
```

#5-3 csplit by specifying the number of lines of multiple files.

```
[jdoe@f33sn1 d5]$ csplit s 10 100 1000
371
4452
51729
635700
[jdoe@f33sn1 d5]$ lh x*
-rw-r--r--. 1 jdoe jdoe  371 Feb  8 11:41 xx00
-rw-r--r--. 1 jdoe jdoe 4.4K Feb  8 11:41 xx01
-rw-r--r--. 1 jdoe jdoe  51K Feb  8 11:41 xx02
-rw-r--r--. 1 jdoe jdoe 621K Feb  8 11:41 xx03

[jdoe@f33sn1 d5]$ cat -n xx00
     1  # /etc/services:
     2  # $Id: services,v 1.49 2017/08/18 12:43:23 ovasik Exp $
     3  #
     4  # Network services, Internet style
     5  # IANA services version: last updated 2016-07-08
     6  #
     7  # Note that it is presently the policy of IANA to assign a single well-known
     8  # port number for both TCP and UDP; hence, most entries here have two entries
     9  # even if the protocol doesn't support UDP operations.
```

#5-4 Try to csplit and see what happens when you try to csplit lines out of range.

```
[jdoe@f33sn1 d5]$ csplit s 10 100 1000 500000
371
4452
51729
csplit: ‘500000’: line number out of range
635700
```

#5-5 Now work with the password file. Using the p file and 'head' command to create a new file p2 with 10 lines.

```
[jdoe@f33sn1 d5]$ lh
total 684K
-rw-r--r--. 1 jdoe jdoe 1.8K Feb  8 11:31 p
-rw-r--r--. 1 jdoe jdoe 677K Feb  8 11:31 s
[jdoe@f33sn1 d5]$ wc -l p
35 p
[jdoe@f33sn1 d5]$ nl p
     1  root:x:0:0:root:/root:/bin/bash
     2  bin:x:1:1:bin:/bin:/sbin/nologin
     3  daemon:x:2:2:daemon:/sbin:/sbin/nologin
     4  adm:x:3:4:adm:/var/adm:/sbin/nologin
     5  lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
     6  sync:x:5:0:sync:/sbin:/bin/sync
     7  shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
     8  halt:x:7:0:halt:/sbin:/sbin/halt
     9  mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
    10  operator:x:11:0:operator:/root:/sbin/nologin
    11  games:x:12:100:games:/usr/games:/sbin/nologin
    12  ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
    13  nobody:x:65534:65534:Kernel Overflow User:/:/sbin/nologin
    14  systemd-coredump:x:999:997:systemd Core Dumper:/:/sbin/nologin
    15  systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin
    16  systemd-resolve:x:193:193:systemd Resolver:/:/sbin/nologin
    17  systemd-timesync:x:998:996:systemd Time Synchronization:/:/sbin/nologin
    18  dbus:x:81:81:System message bus:/:/sbin/nologin
    19  polkitd:x:997:995:User for polkitd:/:/sbin/nologin
    20  rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
    21  cockpit-ws:x:996:991:User for cockpit web service:/nonexisting:/sbin/nologin
    22  cockpit-wsinstance:x:995:990:User for cockpit-ws instances:/nonexisting:/sbin/nologin
    23  tss:x:59:59:Account used for TPM access:/dev/null:/sbin/nologin
    24  setroubleshoot:x:994:989::/var/lib/setroubleshoot:/sbin/nologin
    25  abrt:x:173:173::/etc/abrt:/sbin/nologin
    26  rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
    27  sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
    28  dnsmasq:x:988:988:Dnsmasq DHCP and DNS server:/var/lib/dnsmasq:/usr/sbin/nologin
    29  chrony:x:987:987::/var/lib/chrony:/sbin/nologin
    30  tcpdump:x:72:72::/:/sbin/nologin
    31  jdoe:x:1000:1000::/home/jdoe:/bin/bash
    32  bchoi:x:1001:1001::/home/bchoi:/bin/bash
    33  apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
    34  nginx:x:986:986:Nginx web server:/var/lib/nginx:/sbin/nologin
    35  lcarter:x:1002:0::/home/lcarter:/bin/bash
[jdoe@f33sn1 d5]$ head p > p2
[jdoe@f33sn1 d5]$ nl p2
     1  root:x:0:0:root:/root:/bin/bash
     2  bin:x:1:1:bin:/bin:/sbin/nologin
     3  daemon:x:2:2:daemon:/sbin:/sbin/nologin
     4  adm:x:3:4:adm:/var/adm:/sbin/nologin
     5  lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
     6  sync:x:5:0:sync:/sbin:/bin/sync
     7  shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
     8  halt:x:7:0:halt:/sbin:/sbin/halt
     9  mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
    10  operator:x:11:0:operator:/root:/sbin/nologin

#5-6 csplit the p2 file into two files, each file containing 5 lines.
[jdoe@f33sn1 d5]$ csplit p2 6
183
202
```

```
[jdoe@f33sn1 d5]$ lh x*
-rw-r--r--. 1 jdoe jdoe 183 Feb  8 11:49 xx00
-rw-r--r--. 1 jdoe jdoe 202 Feb  8 11:49 xx01
[jdoe@f33sn1 d5]$ nl xx00
     1  root:x:0:0:root:/root:/bin/bash
     2  bin:x:1:1:bin:/bin:/sbin/nologin
     3  daemon:x:2:2:daemon:/sbin:/sbin/nologin
     4  adm:x:3:4:adm:/var/adm:/sbin/nologin
     5  lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
[jdoe@f33sn1 d5]$ nl xx01
     1  sync:x:5:0:sync:/sbin:/bin/sync
     2  shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
     3  halt:x:7:0:halt:/sbin:/sbin/halt
     4  mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
     5  operator:x:11:0:operator:/root:/sbin/nologin
```

#5-7 To achieve the same result, we can also csplit the file using a string keyword.

```
[jdoe@f33sn1 d5]$ csplit p2 /sync/
183
202
[jdoe@f33sn1 d5]$ nl xx00
     1  root:x:0:0:root:/root:/bin/bash
     2  bin:x:1:1:bin:/bin:/sbin/nologin
     3  daemon:x:2:2:daemon:/sbin:/sbin/nologin
     4  adm:x:3:4:adm:/var/adm:/sbin/nologin
     5  lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
[jdoe@f33sn1 d5]$ nl xx01
     1  sync:x:5:0:sync:/sbin:/bin/sync
     2  shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
     3  halt:x:7:0:halt:/sbin:/sbin/halt
     4  mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
     5  operator:x:11:0:operator:/root:/sbin/nologin
```

#5-8 Pushing the lines up or down based on the string keyword.

```
[jdoe@f33sn1 d5]$ csplit p2 /sync/1
215
170
[jdoe@f33sn1 d5]$ nl xx00
     1  root:x:0:0:root:/root:/bin/bash
     2  bin:x:1:1:bin:/bin:/sbin/nologin
     3  daemon:x:2:2:daemon:/sbin:/sbin/nologin
     4  adm:x:3:4:adm:/var/adm:/sbin/nologin
     5  lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
     6  sync:x:5:0:sync:/sbin:/bin/sync
[jdoe@f33sn1 d5]$ nl xx01
     1  shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
     2  halt:x:7:0:halt:/sbin:/sbin/halt
     3  mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
     4  operator:x:11:0:operator:/root:/sbin/nologin
[jdoe@f33sn1 d5]$ csplit p2 /sync/2
```

```
260
125
[jdoe@f33sn1 d5]$ nl xx00
     1  root:x:0:0:root:/root:/bin/bash
     2  bin:x:1:1:bin:/bin:/sbin/nologin
     3  daemon:x:2:2:daemon:/sbin:/sbin/nologin
     4  adm:x:3:4:adm:/var/adm:/sbin/nologin
     5  lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
     6  sync:x:5:0:sync:/sbin:/bin/sync
     7  shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
[jdoe@f33sn1 d5]$ nl xx01
     1  halt:x:7:0:halt:/sbin:/sbin/halt
     2  mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
     3  operator:x:11:0:operator:/root:/sbin/nologin
[jdoe@f33sn1 d5]$ csplit p2 /sync/3
293
92
[jdoe@f33sn1 d5]$ nl xx00
     1  root:x:0:0:root:/root:/bin/bash
     2  bin:x:1:1:bin:/bin:/sbin/nologin
     3  daemon:x:2:2:daemon:/sbin:/sbin/nologin
     4  adm:x:3:4:adm:/var/adm:/sbin/nologin
     5  lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
     6  sync:x:5:0:sync:/sbin:/bin/sync
     7  shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
     8  halt:x:7:0:halt:/sbin:/sbin/halt
[jdoe@f33sn1 d5]$ nl xx01
     1  mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
     2  operator:x:11:0:operator:/root:/sbin/nologin
[jdoe@f33sn1 d5]$ csplit p2 /sync/-2
105
280
[jdoe@f33sn1 d5]$ nl xx00
     1  root:x:0:0:root:/root:/bin/bash
     2  bin:x:1:1:bin:/bin:/sbin/nologin
     3  daemon:x:2:2:daemon:/sbin:/sbin/nologin
[jdoe@f33sn1 d5]$ nl xx01
     1  adm:x:3:4:adm:/var/adm:/sbin/nologin
     2  lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
     3  sync:x:5:0:sync:/sbin:/bin/sync
     4  shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
     5  halt:x:7:0:halt:/sbin:/sbin/halt
     6  mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
     7  operator:x:11:0:operator:/root:/sbin/nologin
[jdoe@f33sn1 d5]$ lh
total 696K
-rw-r--r--. 1 jdoe jdoe 1.8K Feb  8 11:31 p
-rw-r--r--. 1 jdoe jdoe  385 Feb  8 11:47 p2
-rw-r--r--. 1 jdoe jdoe 677K Feb  8 11:31 s
-rw-r--r--. 1 jdoe jdoe  105 Feb  8 11:55 xx00
-rw-r--r--. 1 jdoe jdoe  280 Feb  8 11:55 xx01
```

```
[jdoe@f33sn1 d5]$ rm -rf xx*
```

#5-9 Add the file name prefix using the -f option.

```
[jdoe@f33sn1 d5]$ csplit -f test_ p2 /sync/-2
105
280
[jdoe@f33sn1 d5]$ lh t*
-rw-r--r--. 1 jdoe jdoe 105 Feb  8 13:49 test_00
-rw-r--r--. 1 jdoe jdoe 280 Feb  8 13:49 test_01
```

#5-10 Use grep and nl commands combination to check the content of a file, s.

```
[jdoe@f33sn1 d5]$ wc -l s
11473 s
[jdoe@f33sn1 d5]$ grep http s
[jdoe@f33sn1 d5]$ grep http s | nl   <<< Check the number of lines containing the word http

[jdoe@f33sn1 d5]$ grep smtp s
smtp            25/tcp          mail
smtp            25/udp          mail
urd             465/tcp         smtps   # URL Rendesvous Directory for SSM / SMTP over SSL (TLS)
rsmtp           2390/tcp                # RSMTP
rsmtp           2390/udp                # RSMTP
[jdoe@f33sn1 d5]$ grep smtp s | nl
     1  smtp            25/tcp          mail
     2  smtp            25/udp          mail
     3  urd             465/tcp         smtps   # URL Rendesvous Directory for SSM / SMTP over SSL (TLS)
     4  rsmtp           2390/tcp                # RSMTP
     5  rsmtp           2390/udp                # RSMTP
```

#5-11 csplit the file at the first instance of smtp line.

```
[jdoe@f33sn1 d5]$ csplit -f test_ s /smtp/
2143
690109
[jdoe@f33sn1 d5]$ lh
total 1.4M
-rw-r--r--. 1 jdoe jdoe 1.8K Feb  8 11:31 p
-rw-r--r--. 1 jdoe jdoe  385 Feb  8 11:47 p2
-rw-r--r--. 1 jdoe jdoe 677K Feb  8 11:31 s
-rw-r--r--. 1 jdoe jdoe 2.1K Feb  8 13:55 test_00
-rw-r--r--. 1 jdoe jdoe 674K Feb  8 13:55 test_01

[jdoe@f33sn1 d5]$ grep smtp test_00                 <<< At this line, it will return no match as there is
no line beginning with smtp.
[jdoe@f33sn1 d5]$ csplit -f test_ s /smtp/1
2180
690072
[jdoe@f33sn1 d5]$ grep smtp test_00
smtp            25/tcp          mail
[jdoe@f33sn1 d5]$ csplit -f test_ s /smtp/2
2217
690035
```

```
[jdoe@f33sn1 d5]$ grep smtp test_00
smtp            25/tcp          mail
smtp            25/udp          mail
```

#5-12 Split the file using a string keyword smtp and check the second file starts with the word, smtp.

```
[jdoe@f33sn1 d5]$ csplit -f test_ s /smtp/
2143
690109
[jdoe@f33sn1 d5]$ grep smtp test_00
[jdoe@f33sn1 d5]$ head test_01
smtp            25/tcp          mail
smtp            25/udp          mail
time            37/tcp          timserver
time            37/udp          timserver
rlp             39/tcp          resource        # resource location
rlp             39/udp          resource        # resource location
nameserver      42/tcp          name            # IEN 116
nameserver      42/udp          name            # IEN 116
nicname         43/tcp          whois
nicname         43/udp          whois
```

#5-13 csplit a file into multiple files, where the string smtp is contained.

```
[jdoe@f33sn1 d5]$ csplit -f test_ s /smtp/ {*}
2143
37
61593
182728
48
445703
[jdoe@f33sn1 d5]$ lh
total 1.4M
-rw-r--r--. 1 jdoe jdoe 1.8K Feb  8 11:31 p
-rw-r--r--. 1 jdoe jdoe  385 Feb  8 11:47 p2
-rw-r--r--. 1 jdoe jdoe 677K Feb  8 11:31 s
-rw-r--r--. 1 jdoe jdoe 2.1K Feb  8 14:16 test_00
-rw-r--r--. 1 jdoe jdoe   37 Feb  8 14:16 test_01
-rw-r--r--. 1 jdoe jdoe  61K Feb  8 14:16 test_02
-rw-r--r--. 1 jdoe jdoe 179K Feb  8 14:16 test_03
-rw-r--r--. 1 jdoe jdoe   48 Feb  8 14:16 test_04
-rw-r--r--. 1 jdoe jdoe 436K Feb  8 14:16 test_05
```

#5-14 We can use the keyword to split the files, use smtp as the keyword.

```
[jdoe@f33sn1 d5]$ grep smtp s
smtp            25/tcp          mail
smtp            25/udp          mail
urd             465/tcp         smtps   # URL Rendesvous Directory for SSM / SMTP over SSL (TLS)
rsmtp           2390/tcp                # RSMTP
rsmtp           2390/udp                # RSMTP

[jdoe@f33sn1 d5]$ csplit -f test_ s /smtp/
2143
```

```
690109
[jdoe@f33sn1 d5]$ ls -lh test*
-rw-r--r--. 1 jdoe jdoe 2.1K Feb  8 19:11 test_00
-rw-r--r--. 1 jdoe jdoe 674K Feb  8 19:11 test_01
[jdoe@f33sn1 d5]$ csplit -f test_ s /smtp/ {1}
2143
37
690072
[jdoe@f33sn1 d5]$ ls -lh test*
-rw-r--r--. 1 jdoe jdoe 2.1K Feb  8 19:13 test_00
-rw-r--r--. 1 jdoe jdoe   37 Feb  8 19:13 test_01
-rw-r--r--. 1 jdoe jdoe 674K Feb  8 19:13 test_02
[jdoe@f33sn1 d5]$ csplit -f test_ s /smtp/ {2}
2143
37
61593
628479
[jdoe@f33sn1 d5]$ ls -lh test*
-rw-r--r--. 1 jdoe jdoe 2.1K Feb  8 19:13 test_00
-rw-r--r--. 1 jdoe jdoe   37 Feb  8 19:13 test_01
-rw-r--r--. 1 jdoe jdoe  61K Feb  8 19:13 test_02
-rw-r--r--. 1 jdoe jdoe 614K Feb  8 19:13 test_03
[jdoe@f33sn1 d5]$ csplit -f test_ s /smtp/ {3}
2143
37
61593
182728
445751
[jdoe@f33sn1 d5]$ ls -lh test*
-rw-r--r--. 1 jdoe jdoe 2.1K Feb  8 19:13 test_00
-rw-r--r--. 1 jdoe jdoe   37 Feb  8 19:13 test_01
-rw-r--r--. 1 jdoe jdoe  61K Feb  8 19:13 test_02
-rw-r--r--. 1 jdoe jdoe 179K Feb  8 19:13 test_03
-rw-r--r--. 1 jdoe jdoe 436K Feb  8 19:13 test_04
[jdoe@f33sn1 d5]$ csplit -f test_ s /smtp/ {4}
2143
37
61593
182728
48
445703
[jdoe@f33sn1 d5]$ ls -lh test*
-rw-r--r--. 1 jdoe jdoe 2.1K Feb  8 19:15 test_00
-rw-r--r--. 1 jdoe jdoe   37 Feb  8 19:15 test_01
-rw-r--r--. 1 jdoe jdoe  61K Feb  8 19:15 test_02
-rw-r--r--. 1 jdoe jdoe 179K Feb  8 19:15 test_03
-rw-r--r--. 1 jdoe jdoe   48 Feb  8 19:15 test_04
-rw-r--r--. 1 jdoe jdoe 436K Feb  8 19:15 test_05
[jdoe@f33sn1 d5]$ csplit -f test_ s /smtp/ {*}
2143
37
```

```
61593
182728
48
445703
[jdoe@f33sn1 d5]$ ls -lh test*
-rw-r--r--. 1 jdoe jdoe 2.1K Feb  8 19:15 test_00
-rw-r--r--. 1 jdoe jdoe   37 Feb  8 19:15 test_01
-rw-r--r--. 1 jdoe jdoe  61K Feb  8 19:15 test_02
-rw-r--r--. 1 jdoe jdoe 179K Feb  8 19:15 test_03
-rw-r--r--. 1 jdoe jdoe   48 Feb  8 19:15 test_04
-rw-r--r--. 1 jdoe jdoe 436K Feb  8 19:15 test_05
[jdoe@f33sn1 d5]$ csplit -f test_ s /smtp/ {5}
2143
37
61593
182728
48
csplit: '/smtp/': match not found on repetition 5
445703
[jdoe@f33sn1 d5]$ ls -lh
total 688K
-rw-r--r--. 1 jdoe jdoe 1.8K Feb  8 11:31 p
-rw-r--r--. 1 jdoe jdoe  385 Feb  8 11:47 p2
-rw-r--r--. 1 jdoe jdoe 677K Feb  8 11:31 s
```

#5-15 Practical 1 - Use the pattern to crate a file with the following content and split the content into multiple files using the string keyword.

```
[jdoe@f33sn1 d5]$ cat > y2
Chapter1
One

Chapter2
Two

Chapter3
Three

Chapter4
Four
[jdoe@f33sn1 d5]$ nl y2
     1  Chapter1
     2  One

     3  Chapter2
     4  Two

     5  Chapter3
     6  Three

     7  Chapter4
     8  Four
```

```
[jdoe@f33sn1 d5]$
[jdoe@f33sn1 d5]$ csplit -f ch_ y2 /Chapter/
0
58
[jdoe@f33sn1 d5]$ nl ch_00
[jdoe@f33sn1 d5]$ csplit -f ch_ y2 /Chapter/1
9
49
[jdoe@f33sn1 d5]$ nl ch_00
     1  Chapter1
[jdoe@f33sn1 d5]$ nl ch_01
     1  One

     2  Chapter2
     3  Two

     4  Chapter3
     5  Three

     6  Chapter4
     7  Four
[jdoe@f33sn1 d5]$
[jdoe@f33sn1 d5]$ csplit -f ch_ y2 /Chapter/ {*}
0
14
14
16
14
[jdoe@f33sn1 d5]$ ll
total 708
-rw-r--r--. 1 jdoe jdoe      0 Feb  8 19:21 ch_00
-rw-r--r--. 1 jdoe jdoe     14 Feb  8 19:21 ch_01
-rw-r--r--. 1 jdoe jdoe     14 Feb  8 19:21 ch_02
-rw-r--r--. 1 jdoe jdoe     16 Feb  8 19:21 ch_03
-rw-r--r--. 1 jdoe jdoe     14 Feb  8 19:21 ch_04
-rw-r--r--. 1 jdoe jdoe   1810 Feb  8 11:31 p
-rw-r--r--. 1 jdoe jdoe    385 Feb  8 11:47 p2
-rw-r--r--. 1 jdoe jdoe 692252 Feb  8 11:31 s
-rw-r--r--. 1 jdoe jdoe     58 Feb  8 19:18 y2
[jdoe@f33sn1 d5]$ nl ch_01
     1  Chapter1
     2  One

[jdoe@f33sn1 d5]$ nl ch_02
     1  Chapter2
     2  Two

[jdoe@f33sn1 d5]$ nl ch_03
     1  Chapter3
     2  Three
```

```
[jdoe@f33sn1 d5]$ nl ch_04
     1  Chapter4
     2  Four

[jdoe@f33sn1 d5]$ csplit -f chapter_ y2 /One/ /Two/ /Three/ /Four/
9
14
14
16
5
[jdoe@f33sn1 d5]$ ls -lh chapter*
-rw-r--r--. 1 jdoe jdoe  9 Feb  8 19:24 chapter_00
-rw-r--r--. 1 jdoe jdoe 14 Feb  8 19:24 chapter_01
-rw-r--r--. 1 jdoe jdoe 14 Feb  8 19:24 chapter_02
-rw-r--r--. 1 jdoe jdoe 16 Feb  8 19:24 chapter_03
-rw-r--r--. 1 jdoe jdoe  5 Feb  8 19:24 chapter_04
[jdoe@f33sn1 d5]$ nl chapter_00
     1  Chapter1
[jdoe@f33sn1 d5]$ nl chapter_01
     1  One

     2  Chapter2
[jdoe@f33sn1 d5]$ nl chapter_02
     1  Two

     2  Chapter3
[jdoe@f33sn1 d5]$ nl chapter_03
     1  Three

     2  Chapter4
[jdoe@f33sn1 d5]$ nl chapter_04
     1  Four
```

Another way to csplit the file.

```
[jdoe@f33sn1 d5]$ csplit -f chapter_ y2 %One%
49
[jdoe@f33sn1 d5]$ ls cha*
chapter_00
[jdoe@f33sn1 d5]$ ls -lh cha*
-rw-r--r--. 1 jdoe jdoe 49 Feb  8 19:26 chapter_00

[jdoe@f33sn1 d5]$ nl chapter_00
     1  One

     2  Chapter2
     3  Two

     4  Chapter3
     5  Three

     6  Chapter4
```

```
   7  Four
```

```
#5-16 Practice 2 - spliting at specific points.
[jdoe@f33sn1 d5]$ wc -l s
11473 s
[jdoe@f33sn1 d5]$ csplit s /http/
547
691705
[jdoe@f33sn1 d5]$ csplit s %http%
691705
[jdoe@f33sn1 d5]$ csplit s %http% {2}   <<<At second occurance
688145
[jdoe@f33sn1 d5]$ csplit s %http% {3}  <<<At third occurance
688067
[jdoe@f33sn1 d5]$ csplit s %http% {20}  <<<At twentieth occurance
601389
[jdoe@f33sn1 d5]$ ls -lh x*
-rw-r--r--. 1 jdoe jdoe 588K Feb  8 19:29 xx00
-rw-r--r--. 1 jdoe jdoe 676K Feb  8 19:28 xx01
```

==

Ex5 Summary

```
csplit -f ch_
csplit k /REGEX/
csplit k /REGEX/ {3}
csplit k %REGEX%
```

==

Exercise 6: echo, printf & yes

time: 15 minutes
==

working with variable in Linux shell

#6-1 variables in Linux shell and printing on the screen using echo command.

```
[jdoe@f33sn1 ~]$ ls
d4  d5
[jdoe@f33sn1 ~]$ pwd
/home/jdoe
[jdoe@f33sn1 ~]$ mkdir d6 && cd d6
[jdoe@f33sn1 d6]$ pwd
/home/jdoe/d6
[jdoe@f33sn1 d6]$ a = 3
-bash: a: command not found
[jdoe@f33sn1 d6]$ a=3
[jdoe@f33sn1 d6]$ b=9
[jdoe@f33sn1 d6]$ c="Australia"
[jdoe@f33sn1 d6]$ echo $a $b $c
3 9 Australia
[jdoe@f33sn1 d6]$ echo "abcd"
abcd
[jdoe@f33sn1 d6]$ echo "abcd" 12345
abcd 12345
[jdoe@f33sn1 d6]$ echo "abcd" 12345 'efgh'
abcd 12345 efgh
```

#6-2 Print username, hostname and a variable

```
[jdoe@f33sn1 d6]$ echo $USER
jdoe
[jdoe@f33sn1 d6]$ whoami
jdoe
[jdoe@f33sn1 d6]$ echo $HOSTNAME
f33sn1
[jdoe@f33sn1 d6]$ echo $xyz

[jdoe@f33sn1 d6]$ xyz=2022
[jdoe@f33sn1 d6]$ echo $xyz
2022
```

#6-3 echo with option -e.

```
[jdoe@f33sn1 d6]$ echo "aa \ bb \n cc"
aa \ bb \n cc
[jdoe@f33sn1 d6]$ echo -e "aa \ bb \n cc"
aa \ bb
 cc
[jdoe@f33sn1 d6]$ echo -e "aa \ bb \n cc \n"
aa \ bb
 cc
```

```
#6-4 Use unicode to print unicode characters.

[jdoe@f33sn1 d6]$ echo -e "\u2665"
♥
[jdoe@f33sn1 d6]$ echo -e "\u2664"
♤
[jdoe@f33sn1 d6]$ echo -e "\u2666"
♦
[jdoe@f33sn1 d6]$ echo -e "\u2667"
♧
[jdoe@f33sn1 d6]$ echo -e "\u30b7\u30c9\u30cb\u30fc"
シドニー
[jdoe@f33sn1 d6]$ echo -e "\u3042\u308a\u304c\u3068\u3046"
ありがとう
```

```
#6-5 "env", environmental variables.
[jdoe@f33sn1 d6]$ env
SHELL=/bin/bash
HISTCONTROL=ignoredups
HISTSIZE=1000
HOSTNAME=f33sn1
EDITOR=/usr/bin/nano
PWD=/home/jdoe/d6
LOGNAME=jdoe
XDG_SESSION_TYPE=tty
[...ommitted for brevity]

[jdoe@f33sn1 d6]$ echo $HISTSIZE
1000
[jdoe@f33sn1 d6]$ echo $TERM
xterm
[jdoe@f33sn1 d6]$ echo $TERMINAL
```

```
# You cannot use "echo --help" to veiw help menu as it is a different type of command. Use "help
echo" instead.
[jdoe@f33sn1 d6]$ echo --help
--help
[jdoe@f33sn1 d6]$ help echo
echo: echo [-neE] [arg ...]
    Write arguments to the standard output.

    Display the ARGs, separated by a single space character and followed by a
    newline, on the standard output.

    Options:
      -n        do not append a newline
      -e        enable interpretation of the following backslash escapes
      -E        explicitly suppress interpretation of backslash escapes
```

```
  `echo' interprets the following backslash-escaped characters:
    \a      alert (bell)
    \b      backspace
    \c      suppress further output
[...ommitted for brevity]

#6-6 Built-in commands
[jdoe@f33sn1 d6]$ #Built-in command
[jdoe@f33sn1 d6]$ compgen -b
.
:
[
alias
bg
bind
break
builtin
caller
cd
command
compgen
complete
compopt
continue
declare
dirs
disown
echo
enable
eval
exec
exit
export
false
fc
fg
getopts
hash
help
history
jobs
kill
let
local
logout
mapfile
popd
printf
pushd
pwd
```

```
read
readarray
readonly
return
set
shift
shopt
source
suspend
test
times
trap
true
type
typeset
ulimit
umask
unalias
unset
wait

[jdoe@f33sn1 d6]$ compgen -b | column
.           compgen     exit        let         return      typeset
:           complete    export      local       set         ulimit
[           compopt     false       logout      shift       umask
alias       continue    fc          mapfile     shopt       unalias
bg          declare     fg          popd        source      unset
bind        dirs        getopts     printf      suspend     wait
break       disown      hash        pushd       test
builtin     echo        help        pwd         times
caller      enable      history     read        trap
cd          eval        jobs        readarray   true
command     exec        kill        readonly    type
```

```
#6-7 Use printf to print different outputs.
[jdoe@f33sn1 d6]$ echo $a $b $c
3 9 Australia
[jdoe@f33sn1 d6]$ printf "%d \t %f \t %s \n" $a $b $c
3       9.000000        Australia
[jdoe@f33sn1 d6]$ printf "%d \t %f \t %s \n" 300 900 "UK"
300     900.000000      UK
[jdoe@f33sn1 d6]$ printf "%d \t %f \t %s \n" 300 900 3.14
300     900.000000      3.14
[jdoe@f33sn1 d6]$ printf "%d \t %f \t %s \n" "dog" 900 3.14
-bash: printf: dog: invalid number
0       900.000000      3.14
[jdoe@f33sn1 d6]$ printf "%d \t %f \t %s \n" 100 900 3.14
100     900.000000      3.14
```

```
#6-8 Use printf to print unicodes
[jdoe@f33sn1 d6]$ printf "\u2665\n"
```

```
♥
[jdoe@f33sn1 d6]$ printf "\u3042\n"
あ
[jdoe@f33sn1 d6]$ printf "\u6771\u4eac\n"
東京

#6-9 Printing decimal (%d), octadecimal (%o) and hexadecial (%x).
[jdoe@f33sn1 d6]$ printf "%d %o %x \n" 15 15 15
15 17 f

[jdoe@f33sn1 d6]$ printf "%x %x %x \n" 10 11 12
a b c

#6-10 Adding spaces to the output.
[jdoe@f33sn1 d6]$ printf "[%10d] \n" 100
[       100]
[jdoe@f33sn1 d6]$ printf "[%20d] \n" 100
[                 100]
[jdoe@f33sn1 d6]$ printf --help    <<<Does not give much help, use man printf
[jdoe@f33sn1 d6]$ man printf
[jdoe@f33sn1 d6]$ man 1 printf
[jdoe@f33sn1 d6]$ man 3 printf
[jdoe@f33sn1 d6]$ man 3 scanf

#6-11 "yes" is an infinite positive.
[jdoe@f33sn1 d6]$ yes "Sydney"
Sydney
Sydney
Sydney
Sydney
Sydney
Sydney
Sydney
Sydney
^C
[jdoe@f33sn1 d6]$ yes "Sydney" | nl
[...ommitted for brevity]
 26416  Sydney
 26417  Sydney
 26418  Sydney
 26419  Sydney
 26420  Sydney
[...ommitted for brevity]

#6-12 Practice and review.

env's variable names are all written in Capital letters.
[jdoe@f33sn1 d6]$ ABC=3000
[jdoe@f33sn1 d6]$ echo $ABC
3000
```

```
[jdoe@f33sn1 d6]$ env
SHELL=/bin/bash
HISTCONTROL=ignoredups
HISTSIZE=1000
HOSTNAME=f33sn1
EDITOR=/usr/bin/nano
PWD=/home/jdoe/d6
LOGNAME=jdoe
XDG_SESSION_TYPE=tty
[...ommitted for brevity]

[jdoe@f33sn1 d6]$ echo $HOME
/home/jdoe
[jdoe@f33sn1 d6]$ echo $TERM
xterm
[jdoe@f33sn1 d6]$ echo $SHELL
/bin/bash

[jdoe@f33sn1 d6]$ echo "aa \t bb"
aa \t bb
[jdoe@f33sn1 d6]$ echo -e  "aa \t bb"
aa       bb

[jdoe@f33sn1 d6]$ printf "%d %d %s\n" 10 20 "best"
10 20 best

[jdoe@f33sn1 d6]$ yes "Spiderman"
[...ommitted for brevity]
Spiderman
Spiderman
Spiderman
Spiderman
Spiderman
Spiderman
Spiderman
Spiderman
Spiderman
Spiderman
Spiderman
Spiderman
Spiderman
^C

[jdoe@f33sn1 d6]$ yes "Spiderman" | nl
[...ommitted for brevity]
179730  Spiderman
179731  Spiderman
179732  Spiderman
179733  Spiderman
179734  Spiderman
179735  Spiderman
```

```
179736  Spiderman
179737  Spiderman
179738  Spiderman
179739  Spiderman
179740  Spiderman
179741  Spiderman
179742  Spiderman
179743  Spiderman
^C
```

==

Ex6 Summary

10. echo
11. printf
12. yes

==

Exercise 7: ls, dir, vdir, & dircolors

time: 20 minutes

```
==============================================================================

#7-1 Create a new directory and change to d7
[jdoe@f33sn1 ~]$ mkdir d7 && cd d7
[jdoe@f33sn1 d7]$ ls
[jdoe@f33sn1 d7]$ mkdir d7a d7b
[jdoe@f33sn1 d7]$ ls
d7a  d7b
[jdoe@f33sn1 d7]$ cal > cal7
[jdoe@f33sn1 d7]$ date > date7
[jdoe@f33sn1 d7]$ ll
total 8
-rw-r--r--. 1 jdoe jdoe 168 Feb  9 21:16 cal7
drwxr-xr-x. 2 jdoe jdoe   6 Feb  9 21:16 d7a
drwxr-xr-x. 2 jdoe jdoe   6 Feb  9 21:16 d7b
-rw-r--r--. 1 jdoe jdoe  30 Feb  9 21:17 date7
[jdoe@f33sn1 d7]$ cat cal7
   February 2022
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28

[jdoe@f33sn1 d7]$ cat date7
Wed 09 Feb 2022 21:17:03 AEDT
[jdoe@f33sn1 d7]$ ls /etc/
abrt                    filesystems    libnl                   passwdqc.conf    services
adjtime                 firewalld      libreport               php.d            sestatus.conf
aliases                 fprintd.conf   libssh                  php-fpm.conf     setroubleshoot
alternatives            fstab          libuser.conf            php-fpm.d        shadow
at.deny                 gcrypt         locale.conf             php.ini          shadow-
audit                   gdbinit        localtime               php-zts.d        shells
authselect              gdbinit.d      login.defs              pinforc          skel
bash_completion.d       gnupg          logrotate.conf          pkcs11           smartmontools
bashrc                  GREP_COLORS    logrotate.d             pkgconfig        sos
bindresvport.blacklist  groff          lvm                     pki              ssh
binfmt.d                group          machine-id              plymouth         ssl
bluetooth               group-         magic                   pm               sssd
chrony.conf             grub2.cfg      mailcap                 polkit-1         statetab.d
chrony.keys             grub.d         makedumpfile.conf.sample  popt.d         subgid
cifs-utils              gshadow        man_db.conf             prelink.conf.d   subgid-
cockpit                 gshadow-       mcelog                  printcap         subuid
[...ommitted for brevity]

#7-2 Study file/directory properties
[jdoe@f33sn1 d7]$ ls d7a
[jdoe@f33sn1 d7]$ # l is for Long
```

```
[jdoe@f33sn1 d7]$ ls -l
total 8
-rw-r--r--. 1 jdoe jdoe 168 Feb  9 21:16 cal7
drwxr-xr-x. 2 jdoe jdoe   6 Feb  9 21:16 d7a
drwxr-xr-x. 2 jdoe jdoe   6 Feb  9 21:16 d7b
-rw-r--r--. 1 jdoe jdoe  30 Feb  9 21:17 date7
```

- or d = file type, - indicates a file, d indicates a directory
rw-r--r-- = indicates privilidges, r=read, w=write, x=execute
2 = hardlink number
joe jdoe = owner and owner group names
168 = file size
Feb 9 21:16 = created date & time

File changed time, there are three different times
Access: access
Modify: change
Change: properties

#7-3 showing all directory properties

```
[jdoe@f33sn1 d7]$ ls -l -a
total 8
drwxr-xr-x. 4 jdoe jdoe  53 Feb  9 21:17 .     <<<self, pointer
drwx------. 6 jdoe jdoe 123 Feb  9 21:15 ..    <<<parent directory
-rw-r--r--. 1 jdoe jdoe 168 Feb  9 21:16 cal7
drwxr-xr-x. 2 jdoe jdoe   6 Feb  9 21:16 d7a
drwxr-xr-x. 2 jdoe jdoe   6 Feb  9 21:16 d7b
-rw-r--r--. 1 jdoe jdoe  30 Feb  9 21:17 date7
```

#7-4 create a hidden file and saved in .

```
[jdoe@f33sn1 d7]$ touch .hiden
[jdoe@f33sn1 d7]$ ls
cal7  d7a  d7b  date7
[jdoe@f33sn1 d7]$ ls -al
total 8
drwxr-xr-x. 4 jdoe jdoe  67 Feb  9 21:37 .
drwx------. 6 jdoe jdoe 123 Feb  9 21:15 ..
-rw-r--r--. 1 jdoe jdoe 168 Feb  9 21:16 cal7
drwxr-xr-x. 2 jdoe jdoe   6 Feb  9 21:16 d7a
drwxr-xr-x. 2 jdoe jdoe   6 Feb  9 21:16 d7b
-rw-r--r--. 1 jdoe jdoe  30 Feb  9 21:17 date7
-rw-r--r--. 1 jdoe jdoe   0 Feb  9 21:37 .hiden
```

#7-5 To hid . and .. directories use -Al

```
[jdoe@f33sn1 d7]$ ls -Al
total 8
-rw-r--r--. 1 jdoe jdoe 168 Feb  9 21:16 cal7
drwxr-xr-x. 2 jdoe jdoe   6 Feb  9 21:16 d7a
drwxr-xr-x. 2 jdoe jdoe   6 Feb  9 21:16 d7b
-rw-r--r--. 1 jdoe jdoe  30 Feb  9 21:17 date7
-rw-r--r--. 1 jdoe jdoe   0 Feb  9 21:37 .hiden
```

#7-6 Viewing file properties and using alias command for command abbreviation

```
[jdoe@f33sn1 d7]$ cp /etc/services s
[jdoe@f33sn1 d7]$ ls
cal7  d7a  d7b  date7  s
[jdoe@f33sn1 d7]$ ls -al
total 688
drwxr-xr-x. 4 jdoe jdoe     76 Feb  9 21:40 .
drwx------. 6 jdoe jdoe    123 Feb  9 21:15 ..
-rw-r--r--. 1 jdoe jdoe    168 Feb  9 21:16 cal7
drwxr-xr-x. 2 jdoe jdoe      6 Feb  9 21:16 d7a
drwxr-xr-x. 2 jdoe jdoe      6 Feb  9 21:16 d7b
-rw-r--r--. 1 jdoe jdoe     30 Feb  9 21:17 date7
-rw-r--r--. 1 jdoe jdoe      0 Feb  9 21:37 .hiden
-rw-r--r--. 1 jdoe jdoe 692252 Feb  9 21:40 s
[jdoe@f33sn1 d7]$ alias ll
alias ll='ls -l --color=auto'
[jdoe@f33sn1 d7]$ ll
total 688
-rw-r--r--. 1 jdoe jdoe    168 Feb  9 21:16 cal7
drwxr-xr-x. 2 jdoe jdoe      6 Feb  9 21:16 d7a
drwxr-xr-x. 2 jdoe jdoe      6 Feb  9 21:16 d7b
-rw-r--r--. 1 jdoe jdoe     30 Feb  9 21:17 date7
-rw-r--r--. 1 jdoe jdoe 692252 Feb  9 21:40 s
[jdoe@f33sn1 d7]$ ll -h
total 688K
-rw-r--r--. 1 jdoe jdoe  168 Feb  9 21:16 cal7
drwxr-xr-x. 2 jdoe jdoe    6 Feb  9 21:16 d7a
drwxr-xr-x. 2 jdoe jdoe    6 Feb  9 21:16 d7b
-rw-r--r--. 1 jdoe jdoe   30 Feb  9 21:17 date7
-rw-r--r--. 1 jdoe jdoe 677K Feb  9 21:40 s
[jdoe@f33sn1 d7]$ alias lh='ls -Alh'
[jdoe@f33sn1 d7]$ lh
total 688K
-rw-r--r--. 1 jdoe jdoe  168 Feb  9 21:16 cal7
drwxr-xr-x. 2 jdoe jdoe    6 Feb  9 21:16 d7a
drwxr-xr-x. 2 jdoe jdoe    6 Feb  9 21:16 d7b
-rw-r--r--. 1 jdoe jdoe   30 Feb  9 21:17 date7
-rw-r--r--. 1 jdoe jdoe    0 Feb  9 21:37 .hiden
-rw-r--r--. 1 jdoe jdoe 677K Feb  9 21:40 s
```

#7-7 use -F option to add "/" to the directory files.

```
[jdoe@f33sn1 d7]$ ll -F
total 688
-rw-r--r--. 1 jdoe jdoe    168 Feb  9 21:16 cal7
drwxr-xr-x. 2 jdoe jdoe      6 Feb  9 21:16 d7a/
drwxr-xr-x. 2 jdoe jdoe      6 Feb  9 21:16 d7b/
-rw-r--r--. 1 jdoe jdoe     30 Feb  9 21:17 date7
-rw-r--r--. 1 jdoe jdoe 692252 Feb  9 21:40 s
[jdoe@f33sn1 d7]$ ls /bin
'['                                    nf-ct-add
```

```
 ab                                nf-ct-events
 abrt                               nf-ct-list
 abrt-action-analyze-backtrace            nf-exp-add
 abrt-action-analyze-c                   nf-exp-delete
[...ommitted for brevity]

#7-8 use -F with * to check executable files under binary (bin) directory
[jdoe@f33sn1 d7]$ ls -F /bin/*
'/bin/['*                                 /bin/nf-ct-add*
 /bin/ab*                                  /bin/nf-ct-events*
 /bin/abrt*                                 /bin/nf-ct-list*
 /bin/abrt-action-analyze-backtrace*            /bin/nf-exp-add*
 /bin/abrt-action-analyze-c*                  /bin/nf-exp-delete*
 /bin/abrt-action-analyze-ccpp-local*           /bin/nf-exp-list*
 /bin/abrt-action-analyze-core*                /bin/nf-log*
 /bin/abrt-action-analyze-oops*                /bin/nf-monitor*
 /bin/abrt-action-analyze-python*               /bin/nf-queue*
 /bin/abrt-action-analyze-vmcore*               /bin/ngettext*
 /bin/abrt-action-analyze-vulnerability*         /bin/nice*
 /bin/abrt-action-analyze-xorg*                /bin/nisdomainname@   <<< notice this? represent a link
file
[...ommitted for brevity]

# Recognizing link files
[jdoe@f33sn1 d7]$ ls -F /bin/yum
/bin/yum@
[jdoe@f33sn1 d7]$ ll -F /bin/yum
lrwxrwxrwx. 1 root root 5 Jul 28  2020 /bin/yum -> dnf-3*
[jdoe@f33sn1 d7]$ ll -F /bin/whois
lrwxrwxrwx. 1 root root 23 Mar 19  2021 /bin/whois -> /etc/alternatives/whois*

#7-9 check a socket file to compare the file properties. "s" at the beginning of output indicates that
the file is a socket file.
[jdoe@f33sn1 d7]$ find / -type s -exec ls -al {} \; 2> /dev/null
srw-rw-rw-. 1 root root 0 Feb  8 09:49 /run/gssproxy.sock
srw-rw----+ 1 root root 0 Feb  8 09:49 /run/php-fpm/www.sock
srwxr-xr-x. 1 root root 0 Feb  8 09:49 /run/mcelog-client
srw-rw-rw-. 1 root root 0 Feb  8 09:49 /run/.heim_org.h5l.kcm-socket
srw-rw-rw-. 1 root root 0 Feb  8 09:49 /run/pcscd/pcscd.comm
srw-rw-rw-. 1 root root 0 Feb  8 09:49 /run/dbus/system_bus_socket
srw-rw-rw-. 1 root root 0 Feb  8 09:49 /run/abrt/abrt.socket
srw-rw-rw-. 1 jdoe jdoe 0 Feb  8 11:25 /run/user/1000/bus
srwxr-xr-x. 1 jdoe jdoe 0 Feb  8 11:25 /run/user/1000/systemd/private
srwxr-xr-x. 1 jdoe jdoe 0 Feb  8 11:25 /run/user/1000/systemd/notify
s---------. 1 jdoe jdoe 0 Feb  8 11:25 /run/user/1000/systemd/inaccessible/sock
srw-------. 1 root root 0 Feb  8 09:49 /run/udev/control
srwxr-xr-x. 1 root root 0 Feb  8 09:49 /run/systemd/home/notify
srw-------. 1 root root 0 Feb  8 09:49 /run/systemd/coredump
srwxrwxrwx. 1 root root 0 Feb  8 09:49 /run/systemd/private
srw-------. 1 root root 0 Feb  8 09:49 /run/systemd/journal/io.systemd.journal
srw-rw-rw-. 1 root root 0 Feb  8 09:49 /run/systemd/journal/socket
```

```
srw-rw-rw-. 1 root root 0 Feb  8 09:49 /run/systemd/journal/stdout
srw-rw-rw-. 1 root root 0 Feb  8 09:49 /run/systemd/journal/dev-log
srw-rw-rw-. 1 root root 0 Feb  8 09:49 /run/systemd/userdb/io.systemd.Home
srw-rw-rw-. 1 root root 0 Feb  8 09:49 /run/systemd/userdb/io.systemd.Multiplexer
srw-rw-rw-. 1 root root 0 Feb  8 09:49 /run/systemd/userdb/io.systemd.DynamicUser
srwxrwxrwx. 1 root root 0 Feb  8 09:49 /run/systemd/notify
s---------. 1 root root 0 Feb  8 09:49 /run/systemd/inaccessible/sock
srw-rw-rw-. 1 root root 0 Feb  8 09:49 /var/lib/gssproxy/default.sock
srw-rw-rw-. 1 root root 0 Feb  8 09:49 /var/lib/sss/pipes/nss

# you can also use -F option and observe that the socket files contains "=" at the end.
[jdoe@f33sn1 d7]$ find / -type s -exec ls -alF {} \; 2> /dev/null
srw-rw-rw-. 1 root root 0 Feb  8 09:49 /run/gssproxy.sock=          <<<Socket files have "=" at the end of the file
srw-rw----+ 1 root root 0 Feb  8 09:49 /run/php-fpm/www.sock=
srwxr-xr-x. 1 root root 0 Feb  8 09:49 /run/mcelog-client=
srw-rw-rw-. 1 root root 0 Feb  8 09:49 /run/.heim_org.h5l.kcm-socket=
srw-rw-rw-. 1 root root 0 Feb  8 09:49 /run/pcscd/pcscd.comm=
srw-rw-rw-. 1 root root 0 Feb  8 09:49 /run/dbus/system_bus_socket=
srw-rw-rw-. 1 root root 0 Feb  8 09:49 /run/abrt/abrt.socket=
srw-rw-rw-. 1 jdoe jdoe 0 Feb  8 11:25 /run/user/1000/bus=
srwxr-xr-x. 1 jdoe jdoe 0 Feb  8 11:25 /run/user/1000/systemd/private=
srwxr-xr-x. 1 jdoe jdoe 0 Feb  8 11:25 /run/user/1000/systemd/notify=
s---------. 1 jdoe jdoe 0 Feb  8 11:25 /run/user/1000/systemd/inaccessible/sock=
srw-------. 1 root root 0 Feb  8 09:49 /run/udev/control=
srwxr-xr-x. 1 root root 0 Feb  8 09:49 /run/systemd/home/notify=
srw-------. 1 root root 0 Feb  8 09:49 /run/systemd/coredump=
srwxrwxrwx. 1 root root 0 Feb  8 09:49 /run/systemd/private=
srw-------. 1 root root 0 Feb  8 09:49 /run/systemd/journal/io.systemd.journal=
srw-rw-rw-. 1 root root 0 Feb  8 09:49 /run/systemd/journal/socket=
srw-rw-rw-. 1 root root 0 Feb  8 09:49 /run/systemd/journal/stdout=
srw-rw-rw-. 1 root root 0 Feb  8 09:49 /run/systemd/journal/dev-log=
srw-rw-rw-. 1 root root 0 Feb  8 09:49 /run/systemd/userdb/io.systemd.Home=
srw-rw-rw-. 1 root root 0 Feb  8 09:49 /run/systemd/userdb/io.systemd.Multiplexer=
srw-rw-rw-. 1 root root 0 Feb  8 09:49 /run/systemd/userdb/io.systemd.DynamicUser=
srwxrwxrwx. 1 root root 0 Feb  8 09:49 /run/systemd/notify=
s---------. 1 root root 0 Feb  8 09:49 /run/systemd/inaccessible/sock=
srw-rw-rw-. 1 root root 0 Feb  8 09:49 /var/lib/gssproxy/default.sock=
srw-rw-rw-. 1 root root 0 Feb  8 09:49 /var/lib/sss/pipes/nss=


#7-10 p stand for pipe, try the following command.
[jdoe@f33sn1 d7]$ find / -type p -exec ls -alF {} \; 2> /dev/null
prw-------. 1 root root 0 Feb  8 09:49 /run/initctl|
prw-------. 1 root root 0 Feb  8 09:49 /run/dmeventd-client|
prw-------. 1 root root 0 Feb  8 09:49 /run/dmeventd-server|
p---------. 1 jdoe jdoe 0 Feb  8 11:25 /run/user/1000/systemd/inaccessible/fifo|
prw-------. 1 root root 0 Feb  8 09:49 /run/systemd/inhibit/2.ref|
prw-------. 1 root root 0 Feb  8 09:49 /run/systemd/inhibit/1.ref|
prw-------. 1 root root 0 Feb  9 13:05 /run/systemd/sessions/6.ref|
prw-------. 1 root root 0 Feb  8 11:25 /run/systemd/sessions/1.ref|
p---------. 1 root root 0 Feb  8 09:49 /run/systemd/inaccessible/fifo|
prw-------. 1 root root 0 Feb  8 09:49 /var/lib/nfs/rpc_pipefs/gssd/clntXX/gssd|
```

#7-11 -i option displays the inode of files, inodes are unique file numbers. Book has an isbn, inode is the same concept to isbn for books.

```
[jdoe@f33sn1 d7]$ ls -i
 4670213 cal7   8636411 d7a  14477925 d7b   4670214 date7   4670216 s
[jdoe@f33sn1 d7]$ lh -i
total 688K
 4670213 -rw-r--r--. 1 jdoe jdoe  168 Feb  9 21:16 cal7
 8636411 drwxr-xr-x. 2 jdoe jdoe    6 Feb  9 21:16 d7a
14477925 drwxr-xr-x. 2 jdoe jdoe    6 Feb  9 21:16 d7b
 4670214 -rw-r--r--. 1 jdoe jdoe   30 Feb  9 21:17 date7
 4670215 -rw-r--r--. 1 jdoe jdoe    0 Feb  9 21:37 .hiden
 4670216 -rw-r--r--. 1 jdoe jdoe 677K Feb  9 21:40 s
```

#7-12 Use -R (recursive) option to go inside each directory and view the files.

```
[jdoe@f33sn1 d7]$ ls /etc/

[jdoe@f33sn1 d7]$ ls -R /etc/
```

#7-13 -S option for size.

```
[jdoe@f33sn1 d7]$ ls -Sl /etc
total 1360
-rw-r--r--.  1 root root    692252 Jun 23  2020 services       <<< biggest file, decending order from large to small size
-rw-r--r--.  1 root root     77941 Jul 29  2020 jwhois.conf
-rw-r--r--.  1 root root     67454 Apr 22  2020 mime.types
-rw-r--r--.  1 root root     63599 Mar  2  2021 php.ini
-rw-r--r--.  1 root dnsmasq  27839 Oct  1  2020 dnsmasq.conf
-rw-r--r--.  1 root root     27500 Mar 22  2021 ld.so.cache
-rw-r--r--.  1 root root     10315 Oct 19  2020 nanorc
-rw-r--r--.  1 root root      8493 Mar 19  2021 kdump.conf
-rw-r--r--.  1 root root      7504 Jul 30  2020 login.defs
-rw-r--r--.  1 root root      6568 Jun 23  2020 protocols
-rw-r--r--.  1 root root      5799 Sep  9  2020 idmapd.conf
-rw-r--r--.  1 root root      5235 Sep 16  2020 man_db.conf
-rw-r--r--.  1 root root      5122 Aug  4  2020 makedumpfile.conf.sample
-rw-r--r--.  1 root root      4937 Mar  2  2021 php-fpm.conf
[...ommitted for brevity]

# Add "r" to reverse the order. This time smallest file to the largest in descending order.
[jdoe@f33sn1 d7]$ ls -Slr /etc
total 1360
-rw-r--r--.  1 root root         0 Jun 23  2020 motd
-rw-r--r--.  1 root root         0 Jun 23  2020 exports
-rw-r--r--.  1 root root         0 Jul 30  2020 environment
-rw-------.  1 root root         0 Mar 19  2021 crypttab
-rw-r--r--.  1 root root         1 Jul 27  2020 at.deny
```

```
drwxr-xr-x.  2 root root       6 Oct  2  2020 tmpfiles.d
drwxr-xr-x.  2 root root       6 Jul 29  2020 terminfo
drwxr-xr-x.  2 root root       6 Jul 28  2020 statetab.d
drwxr-xr-x.  2 root root       6 Jul 28  2020 sasl2
drwxr-xr-x.  2 root root       6 Sep 18  2020 rsyslog.d
[...ommitted for brevity]

#7-14 Use ls command with '-t" option to display files in time order.
[jdoe@f33sn1 d7]$ ls -Slt /etc
total 1360
----------.  1 root root    1222 Feb  7 15:09 shadow
drwxr-x---.  2 root root      21 Mar 23  2021 sudoers.d
----------.  1 root root     619 Mar 23  2021 gshadow
-rw-r--r--.  1 root root     768 Mar 23  2021 group
-rw-r--r--.  1 root root      58 Mar 23  2021 subgid
-rw-r--r--.  1 root root      58 Mar 23  2021 subuid
-rw-r--r--.  1 root root    1810 Mar 23  2021 passwd
-rw-r--r--.  1 root root   27500 Mar 22  2021 ld.so.cache
[...ommitted for brevity]

# Reverse the order
[jdoe@f33sn1 d7]$ ls -Sltr /etc
total 1360
-rw-r--r--.  1 root root      99 Mar 14  2010 passwdqc.conf
-rw-r--r--.  1 root root    2391 Jul 24  2015 libuser.conf
-rw-r-----.  1 root root     191 Nov  5  2019 libaudit.conf
-rw-r--r--.  1 root root      20 Feb 11  2020 fprintd.conf
-rw-r--r--.  1 root root   67454 Apr 22  2020 mime.types
-rw-r--r--.  1 root root     272 Apr 22  2020 mailcap
-rw-r--r--.  1 root root     496 Jun  8  2020 logrotate.conf
-rw-r--r--.  1 root root      44 Jun 23  2020 shells
-rw-r--r--.  1 root root  692252 Jun 23  2020 services
-rw-r--r--.  1 root root    6568 Jun 23  2020 protocols
-rw-r--r--.  1 root root    1816 Jun 23  2020 profile
[...ommitted for brevity]

#7-15 Getting help for ls
[jdoe@f33sn1 d7]$ ls --help

#7-16 To view large list of files one item on each line
[jdoe@f33sn1 d7]$ ls /etc -1
abrt
adjtime
aliases
alternatives
at.deny
audit
authselect
[...ommitted for brevity]

#7-17 dir, vdir and showing files with colours
```

```
[jdoe@f33sn1 d7]$ ls
cal7  d7a  d7b  date7  s
[jdoe@f33sn1 d7]$ dir
cal7  d7a  d7b  date7  s
[jdoe@f33sn1 d7]$ vdir
total 688
-rw-r--r--. 1 jdoe jdoe    168 Feb  9 21:16 cal7
drwxr-xr-x. 2 jdoe jdoe      6 Feb  9 21:16 d7a
drwxr-xr-x. 2 jdoe jdoe      6 Feb  9 21:16 d7b
-rw-r--r--. 1 jdoe jdoe     30 Feb  9 21:17 date7
-rw-r--r--. 1 jdoe jdoe 692252 Feb  9 21:40 s
[jdoe@f33sn1 d7]$ whereis dir
dir: /usr/bin/dir /usr/share/man/man1/dir.1.gz /usr/share/info/dir
[jdoe@f33sn1 d7]$ whereis vdir
vdir: /usr/bin/vdir /usr/share/man/man1/vdir.1.gz
[jdoe@f33sn1 d7]$ ll /usr/bin/dir
-rwxr-xr-x. 1 root root 141872 Aug 17  2020 /usr/bin/dir
[jdoe@f33sn1 d7]$ dir --color=auto
cal7  d7a  d7b  date7  s
[jdoe@f33sn1 d7]$ vdir --color=tty
total 688
-rw-r--r--. 1 jdoe jdoe    168 Feb  9 21:16 cal7
drwxr-xr-x. 2 jdoe jdoe      6 Feb  9 21:16 d7a
drwxr-xr-x. 2 jdoe jdoe      6 Feb  9 21:16 d7b
-rw-r--r--. 1 jdoe jdoe     30 Feb  9 21:17 date7
-rw-r--r--. 1 jdoe jdoe 692252 Feb  9 21:40 s
[jdoe@f33sn1 d7]$ dir --color=tty
cal7  d7a  d7b  date7  s
[jdoe@f33sn1 d7]$ dir /etc --color=tty
abrt                     filesystems    libnl                    passwdqc.conf    services
adjtime                  firewalld      libreport                php.d            sestatus.conf
aliases                  fprintd.conf   libssh                   php-fpm.conf     setroubleshoot
alternatives             fstab          libuser.conf             php-fpm.d        shadow
at.deny                  gcrypt         locale.conf              php.ini          shadow-
audit                    gdbinit        localtime                php-zts.d        shells
authselect               gdbinit.d      login.defs               pinforc          skel
bash_completion.d        gnupg          logrotate.conf           pkcs11           smartmontools
bashrc                   GREP_COLORS    logrotate.d              pkgconfig        sos
bindresvport.blacklist   groff          lvm                      pki              ssh
binfmt.d                 group          machine-id               plymouth         ssl
bluetooth                group-         magic                    pm               sssd
chrony.conf              grub2.cfg      mailcap                  polkit-1         statetab.d
chrony.keys              grub.d         makedumpfile.conf.sample popt.d           subgid
cifs-utils               gshadow        man_db.conf              prelink.conf.d   subgid-

#7-18 ls is an alias, using "\" infront of ls will remove the options.
[jdoe@f33sn1 d7]$ alias ls
alias ls='ls --color=auto'
[jdoe@f33sn1 d7]$ \ls
cal7  d7a  d7b  date7  s
[jdoe@f33sn1 d7]$ \ls --color=auto
```

```
cal7  d7a  d7b  date7  s
```

#7-19 Checking alias

```
[jdoe@f33sn1 d7]$ ll
total 688
-rw-r--r--. 1 jdoe jdoe    168 Feb  9 21:16 cal7
drwxr-xr-x. 2 jdoe jdoe      6 Feb  9 21:16 d7a
drwxr-xr-x. 2 jdoe jdoe      6 Feb  9 21:16 d7b
-rw-r--r--. 1 jdoe jdoe     30 Feb  9 21:17 date7
-rw-r--r--. 1 jdoe jdoe 692252 Feb  9 21:40 s
[jdoe@f33sn1 d7]$ l.
.  ..  .hiden
[jdoe@f33sn1 d7]$ alias ll
alias ll='ls -l --color=auto'
[jdoe@f33sn1 d7]$ alias l.
alias l.='ls -d .* --color=auto'
[jdoe@f33sn1 d7]$ alias ls
alias ls='ls --color=auto'
[jdoe@f33sn1 d7]$ alias
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias l.='ls -d .* --color=auto'
alias lh='ls -Alh'
alias ll='ls -l --color=auto'
alias ls='ls --color=auto'
alias which='(alias; declare -f) | /usr/bin/which --tty-only --read-alias --read-functions --show-tilde --
show-dot'
alias xzegrep='xzegrep --color=auto'
alias xzfgrep='xzfgrep --color=auto'
alias xzgrep='xzgrep --color=auto'
alias zegrep='zegrep --color=auto'
alias zfgrep='zfgrep --color=auto'
alias zgrep='zgrep --color=auto'
```

#7-20 Checking directory colors

```
[jdoe@f33sn1 d7]$ dircolors
LS_COLORS='rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:
or=40;31;01:mi=00:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31
:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.taz=01;31:*.lha=01;31:*.lz4=01;31:*.lzh=01;31:*.lzma=01;31:
*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=
01;31:*.lz=01;31:*.lzo=01;31:*.xz=01;31:*.zst=01;31:*.tzst=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;
31:*.tbz2=01;31:*.tz=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sar=0
1;31:*.rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.cab=01
;31:*.wim=01;31:*.swm=01;31:*.dwm=01;31:*.esd=01;31:*.jpg=01;35:*.jpeg=01;35:*.mjpg=01;35:*
.mjpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm
=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.
pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.webm=01;35:*.w
ebp=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=0
1;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=0
1;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.ogv=0
```

```
1;35:*.ogx=01;35:*.aac=00;36:*.au=00;36:*.flac=00;36:*.m4a=00;36:*.mid=00;36:*.midi=00;36:*.m
ka=00;36:*.mp3=00;36:*.mpc=00;36:*.ogg=00;36:*.ra=00;36:*.wav=00;36:*.oga=00;36:*.opus=00;3
6:*.spx=00;36:*.xspf=00;36:';
export LS_COLORS
```

The Directory colors are defined in the following directory.

```
[jdoe@f33sn1 d7]$ nl /etc/DIR_COLORS
     1  # This file goes in the /etc directory, and must be world readable.
     2  # You can copy this file to .dir_colors in your $HOME directory to override
     3  # the system defaults.

     4  # Configuration file for dircolors, a utility to help you set the
     5  # LS_COLORS environment variable used by GNU ls with the --color option.

     6  # Copyright (C) 1996-2020 Free Software Foundation, Inc.
     7  # Copying and distribution of this file, with or without modification,
     8  # are permitted provided the copyright notice and this notice are preserved.

     9  # The keywords COLOR, OPTIONS, and EIGHTBIT (honored by the
    10  # slackware version of dircolors) are recognized but ignored.

    11  # For compatibility, the pattern "^COLOR.*none" is recognized as a way to
    12  # disable colorization.  See https://bugzilla.redhat.com/1349579 for details.

    13  # Below are TERM entries, which can be a glob patterns, to match
    14  # against the TERM environment variable to determine if it is colorizable.
    15  TERM Eterm
    16  TERM ansi
    17  TERM *color*
    18  TERM con[0-9]*x[0-9]*
    19  TERM cons25
    20  TERM console
[...ommmited for brevity]
```

* You cannot change this color as a user.
* Only if you are using a Linux VM with multiuser console access.

#7-21 Customizing the color

```
[jdoe@f33sn1 d7]$ dircolors --help
Usage: dircolors [OPTION]... [FILE]
Output commands to set the LS_COLORS environment variable.

Determine format of output:
  -b, --sh, --bourne-shell    output Bourne shell code to set LS_COLORS
  -c, --csh, --c-shell        output C shell code to set LS_COLORS
  -p, --print-database        output defaults
      --help     display this help and exit
      --version  output version information and exit

If FILE is specified, read it to determine which colors to use for which
file types and extensions.  Otherwise, a precompiled database is used.
```

```
For details on the format of these files, run 'dircolors --print-database'.

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Full documentation <https://www.gnu.org/software/coreutils/dircolors>
or available locally via: info '(coreutils) dircolors invocation'

[jdoe@f33sn1 d7]$ dircolors -p > ~/.dir_colors
[jdoe@f33sn1 d7]$ vi ~/.dir_colors
```

Change DIR from 34(Blue) background to 31(Red)
DIR 01;34 # directory ===> DIR 01;31 # directory

Open another terminal window and check the changes, the filenames will be displayed in red.

```
[jdoe@f33sn1 ~]$ ls
d4  d5  d6  d7
```

==

Ex7 Summary

```
ls
-a all
-A exclude . .. all
-l long, detailed info
-h human-readable
-F file properties (one of */=>@|)
-i inode
-R recursive

-S size
-r reverse
-t time
-t modification time
        -u access time
        -c change -time
```

==

Exercise 8: paste & join

time: 20 minutes

==

Paste and join can replace excel.
You have to understand the concept of field-column & record-row.

name	math	eng
Peter	85	93
Diana	93	87
Tony	90	95
Lara	99	98
Bruce	89	80

#8-1 Make a directory d8 and change the working directory. Create three files with names, math & eng test scores.

```
[jdoe@f33sn1 ~]$ mkdir d8 && cd d8
[jdoe@f33sn1 d8]$ cat > name
Peter
Diana
Tony
Lara
Bruce           CTRL+D to save and exit
[jdoe@f33sn1 d8]$ nl name
     1  Peter
     2  Diana
     3  Tony
     4  Lara
     5  Bruce
[jdoe@f33sn1 d8]$ echo -e "85\n93\n90\n99\n89" > math
[jdoe@f33sn1 d8]$ nl math
     1  85
     2  93
     3  90
     4  99
     5  89
[jdoe@f33sn1 d8]$  cat > eng
93
87
95
98
80
[jdoe@f33sn1 d8]$ nl eng
     1  93
     2  87
     3  95
     4  98
     5  80
```

#8-2 Print content of all three files using the "cat" command. Use past command to print name in rows and columns.

```
[jdoe@f33sn1 d8]$ cat name math eng
Peter
Diana
Tony
Lara
Bruce
85
93
90
99
89
93
87
95
98
80
[jdoe@f33sn1 d8]$ paste name math eng
Peter   85      93
Diana   93      87
Tony    90      95
Lara    99      98
Bruce   89      80
```

#8-3 You already know how to save the paste output using the >.

```
[jdoe@f33sn1 d8]$ paste name math eng > result1
[jdoe@f33sn1 d8]$ nl result1
     1  Peter   85      93
     2  Diana   93      87
     3  Tony    90      95
     4  Lara    99      98
     5  Bruce   89      80
```

#8-4 Changing the orientation of the output using "-s" option

```
[jdoe@f33sn1 d8]$ paste name math eng -s
Peter   Diana   Tony    Lara    Bruce
85      93      90      99      89
93      87      95      98      80
[jdoe@f33sn1 d8]$ paste name math eng -s > result2
[jdoe@f33sn1 d8]$ nl result2
     1  Peter   Diana   Tony    Lara    Bruce
     2  85      93      90      99      89
     3  93      87      95      98      80
[jdoe@f33sn1 d8]$ ls
eng  math  name  result1  result2
[jdoe@f33sn1 d8]$ ll
total 20
-rw-r--r--. 1 jdoe jdoe 15 Feb 11 16:11 eng
-rw-r--r--. 1 jdoe jdoe 15 Feb 11 16:10 math
-rw-r--r--. 1 jdoe jdoe 28 Feb 11 16:26 name
```

```
-rw-r--r--. 1 jdoe jdoe 58 Feb 11 16:31 result1
-rw-r--r--. 1 jdoe jdoe 58 Feb 11 16:31 result2
```

#8-5 paste with seperator.

```
[jdoe@f33sn1 d8]$ paste name
Peter
Diana
Tony
Lara
Bruce
[jdoe@f33sn1 d8]$ paste name -s
Peter   Diana   Tony    Lara    Bruce
[jdoe@f33sn1 d8]$ paste name -s -d"="
Peter=Diana=Tony=Lara=Bruce
[jdoe@f33sn1 d8]$ paste name -s -d"#"
Peter#Diana#Tony#Lara#Bruce
[jdoe@f33sn1 d8]$ paste name -s -d"@"
Peter@Diana@Tony@Lara@Bruce
[jdoe@f33sn1 d8]$ paste name -s -d"|"
Peter|Diana|Tony|Lara|Bruce
[jdoe@f33sn1 d8]$ paste name -s -d","
Peter,Diana,Tony,Lara,Bruce
[jdoe@f33sn1 d8]$ paste name -s -d"-,=$"
Peter-Diana,Tony=Lara$Bruce
[jdoe@f33sn1 d8]$ paste name -s -d"#$"
Peter#Diana$Tony#Lara$Bruce
[jdoe@f33sn1 d8]$ paste name -s -d"-"
Peter-Diana-Tony-Lara-Bruce
```

#8-6 Create result using name & math, name & eng. Use join to use the name as the common values in union. Check the output difference between paste and join.

```
[jdoe@f33sn1 d8]$ paste name math
Peter   85
Diana   93
Tony    90
Lara    99
Bruce   89
[jdoe@f33sn1 d8]$ paste name math > m1
[jdoe@f33sn1 ex8]$ paste name eng > e1
[jdoe@f33sn1 d8]$ paste name eng
Peter   93
Diana   87
Tony    95
Lara    98
Bruce   80
[jdoe@f33sn1 d8]$ paste m1 e1
Peter   85      Peter   93
Diana   93      Diana   87
Tony    90      Tony    95
Lara    99      Lara    98
Bruce   89      Bruce   80
```

```
[jdoe@f33sn1 d8]$ join m1 e1
Peter 85 93
Diana 93 87
Tony 90 95
Lara 99 98
Bruce 89 80
```

#8-7 What if some students were absent and did not take the exam. Remove Diana's math result and Lara's English result.

```
[jdoe@f33sn1 d8]$ nano m1
[jdoe@f33sn1 d8]$ cat m1
Peter   85
Diana
Tony    90
Lara    99
Bruce   89
[jdoe@f33sn1 d8]$ nano e1
[jdoe@f33sn1 d8]$ cat e1
Peter   93
Diana   87
Tony    95
Lara
Bruce   80
[jdoe@f33sn1 d8]$ paste m1 e1
Peter   85      Peter   93
Diana   Diana   87
Tony    90      Tony    95
Lara    99      Lara
Bruce   89      Bruce   80
[jdoe@f33sn1 d8]$ join m1 e1
Peter 85 93
Diana 87
Tony 90 95
Lara 99
Bruce 89 80
[jdoe@f33sn1 d8]$ join -a 1 m1 e1
Peter 85 93
Diana 87
Tony 90 95
Lara 99
Bruce 89 80
[jdoe@f33sn1 d8]$ join -a 2 m1 e1
Peter 85 93
Diana 87
Tony 90 95
Lara 99
Bruce 89 80
[jdoe@f33sn1 d8]$ join -a 1 -a 2 m1 e1
Peter 85 93
Diana 87
Tony 90 95
```

```
Lara 99
Bruce 89 80
[jdoe@f33sn1 d8]$ join -e"00" -o 0,1.2,2.1 m1 e1
Peter 85 Peter
Diana 00 Diana
Tony 90 Tony
Lara 99 Lara
Bruce 89 Bruce
[jdoe@f33sn1 d8]$ join -e"00" -o 0,1.2,2.2 m1 e1
Peter 85 93
Diana 00 87
Tony 90 95
Lara 99 00
Bruce 89 80
[jdoe@f33sn1 d8]$ join -e"00" -o 0,1.2,2.2 m1 e1 > result3
[jdoe@f33sn1 d8]$ nl result3
     1  Peter 85 93
     2  Diana 00 87
     3  Tony 90 95
     4  Lara 99 00
     5  Bruce 89 80
```

```
#8-8 Use of option "-v", finds value of empty common field
[jdoe@f33sn1 d8]$ nano m1
[jdoe@f33sn1 d8]$ cat m1
Peter   85
Diana
Tony    90
Lara    99
Bruce   89
        100
[jdoe@f33sn1 d8]$ join -v 1 -o 0,1.2,2.2 m1 e1
100
[jdoe@f33sn1 d8]$ join -v 2 -o 0,1.2,2.2 m1 e1
```

```
column

name   math    eng     field
--------------------
Peter   85      93     row
Diana   93      87
Tony    90      95     (records)
Lara    99      98
Bruce   89      80
```

csv concept.

===

Ex8 Summary

Review this command again.

join -e"00" -o 0,1.2,2.2 m1 e1

-e"00"   - Fills empty records (n/a in excel)
-o 0,1.2, 2.2 m1 e1 - records from column 0, column 2 of file 1 (m1), column 2 of file 2 (e1)

paste - merge lines of files
    paste [OPTION]... [FILE]...

-d, --delimiters=LIST, reuse characters from LIST instead of TABs
-s, --serial, paste one file at a time instead of in parallel.

paste [option]... [FILE]...

----

join - join lines of two files on a common field
join [option]... FILE1 FILE2
join --help

===

Exercise 9: cut & awk

time: 20 minutes
==

Remember the row and columns concept from the previous exercise?

```
name   math    eng
--------------------
Peter   85     93
Diana   93     87
Tony    90     95
Lara    99     98
Bruce   89     80
```

----
* Before going into exercises. Give a brief overview using the summary.

#9-1 Use files from previous exercise 8. First, create a directory named d9 and copy the all content of d8, using the 'cp' command.

```
[jdoe@f33sn1 ~]$ ls
d1 d2 d3 d4  d5  d6  d7  d8
[jdoe@f33sn1 ~]$ ls d8
e1  eng  m1  math  name  result1  result2  result3
[jdoe@f33sn1 ~]$ mkdir d9
[jdoe@f33sn1 ~]$ cp d8/* d9
[jdoe@f33sn1 ~]$ cd d9
[jdoe@f33sn1 d9]$ ls
e1  eng  m1  math  name  result1  result2  result3
[jdoe@f33sn1 d9]$ nl name
     1  Peter
     2  Diana
     3  Tony
     4  Lara
     5  Bruce
[jdoe@f33sn1 d9]$ nl math
     1  85
     2  93
     3  90
     4  99
     5  89
[jdoe@f33sn1 d9]$ nl eng
     1  93
     2  87
     3  95
     4  98
     5  80
```

#9-2 Add "| tee r" at the end of the paste command to display the content as well as save the content to file r.

```
[jdoe@f33sn1 d9]$ paste name eng math | tee r
```

```
Peter   93      85
Diana   87      93
Tony    95      90
Lara    98      99
Bruce   80      89
[jdoe@f33sn1 d9]$ cat r
Peter   93      85
Diana   87      93
Tony    95      90
Lara    98      99
Bruce   80      89
[jdoe@f33sn1 d9]$ nl r
     1  Peter   93      85
     2  Diana   87      93
     3  Tony    95      90
     4  Lara    98      99
     5  Bruce   80      89
```

#9-3 Use the '| cut' command to get a specific item in the field in a string. The following example is to get each item from the date command output.

```
[jdoe@f33sn1 d9]$ date
Fri 11 Feb 2022 19:53:29 AEDT
[jdoe@f33sn1 d9]$
[jdoe@f33sn1 d9]$ date | cut -f1
Fri 11 Feb 2022 19:54:26 AEDT
[jdoe@f33sn1 d9]$ date | cut -f2
Fri 11 Feb 2022 19:54:43 AEDT
[jdoe@f33sn1 d9]$ date | cut -f3
Fri 11 Feb 2022 19:54:45 AEDT
[jdoe@f33sn1 d9]$ date | cut -f4
Fri 11 Feb 2022 19:54:47 AEDT
[jdoe@f33sn1 d9]$ date | cut -f5
Fri 11 Feb 2022 19:56:11 AEDT
[jdoe@f33sn1 d9]$ date | cut -f6
Fri 11 Feb 2022 19:56:14 AEDT
[jdoe@f33sn1 d9]$ date | cut -d' ' -f1
Fri
[jdoe@f33sn1 d9]$ date | cut -d' ' -f1
Fri
[jdoe@f33sn1 d9]$ date | cut -d' ' -f2
11
[jdoe@f33sn1 d9]$ date | cut -d' ' -f3
Feb
[jdoe@f33sn1 d9]$ date | cut -d' ' -f4
2022
[jdoe@f33sn1 d9]$ date | cut -d' ' -f5
19:55:46
[jdoe@f33sn1 d9]$ date | cut -d' ' -f6
AEDT

[jdoe@f33sn1 d9]$ who
```

```
jdoe     pts/0        2022-02-11 19:08 (192.168.56.1)
[jdoe@f33sn1 d9]$ who | cut -d' ' -f1
jdoe
[jdoe@f33sn1 d9]$ who | cut -d' ' -f6
pts/0
[jdoe@f33sn1 d9]$ who | cut -d' ' -f14
2022-02-11
[jdoe@f33sn1 d9]$ who | cut -d' ' -f15
19:08
[jdoe@f33sn1 d9]$ who | cut -d' ' -f16
(192.168.56.1)
```

#9-4 Use "echo {A..Z}" to print a set of alphabets or a number range.

```
[jdoe@f33sn1 d9]$ echo "ABC..Z"
ABC..Z
[jdoe@f33sn1 d9]$ echo {A..Z}
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
[jdoe@f33sn1 d9]$ echo {A..M}
A B C D E F G H I J K L M
[jdoe@f33sn1 d9]$ echo {a..z}
a b c d e f g h i j k l m n o p q r s t u v w x y z
[jdoe@f33sn1 d9]$ echo {a..m}
a b c d e f g h i j k l m
[jdoe@f33sn1 d9]$ echo {A..z}
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [  ] ^ _ ` a b c d e f g h i j k l m n o p q r s t u v w x y
z
[jdoe@f33sn1 d9]$ echo {1..10}
1 2 3 4 5 6 7 8 9 10
[jdoe@f33sn1 d9]$ echo {1..20}
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
[jdoe@f33sn1 d9]$ echo {1..30}
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
```

#9-5 Trim the spaces to output as continuous strings.

```
[jdoe@f33sn1 d9]$ echo {A..Z} | tr -d ' '
ABCDEFGHIJKLMNOPQRSTUVWXYZ
[jdoe@f33sn1 d9]$ echo {1..20} | tr -d ' '
1234567891011121314151617181920
```

*Try 'tr --help' for more information.

#9-6 Combine "| cut -c 1" to call by character positions.

```
[jdoe@f33sn1 d9]$ echo {A..Z}
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
[jdoe@f33sn1 d9]$ echo {A..Z} | tr -d ' ' | cut -c 1
A
[jdoe@f33sn1 d9]$ echo {A..Z} | tr -d ' ' | cut -c 2
B
[jdoe@f33sn1 d9]$ echo {A..Z} | tr -d ' ' | cut -c 3
C
[jdoe@f33sn1 d9]$ echo {A..Z} | tr -d ' ' | cut -c 10
```

```
J
[jdoe@f33sn1 d9]$ echo {A..Z} | tr -d ' ' | cut -c 20
T

# Call multiple items by using a comma separator after the -c option.
[jdoe@f33sn1 d9]$ echo {A..Z} | tr -d ' ' | cut -c 1,2,3
ABC
[jdoe@f33sn1 d9]$ echo {A..Z} | tr -d ' ' | cut -c 1,10,20
AJT
[jdoe@f33sn1 d9]$ echo {A..Z} | tr -d ' ' | cut -c 20,1     <<<Cannot reverse the output
AT

# Using 'range' to output strings.
[jdoe@f33sn1 d9]$ echo {A..Z} | tr -d ' ' | cut -c 1-10
ABCDEFGHIJ
[jdoe@f33sn1 d9]$ echo {A..Z} | tr -d ' ' | cut -c 20-26
TUVWXYZ
[jdoe@f33sn1 d9]$ echo {A..Z} | tr -d ' ' | cut -c 26-1
cut: invalid decreasing range
Try 'cut --help' for more information.                    <<<Cannot reverse the output

# Use the default range. Same as Python string methods.
[jdoe@f33sn1 d9]$ echo {A..Z} | tr -d ' ' | cut -c -5
ABCDE
[jdoe@f33sn1 d9]$ echo {A..Z} | tr -d ' ' | cut -c -10
ABCDEFGHIJ
[jdoe@f33sn1 d9]$ echo {A..Z} | tr -d ' ' | cut -c 10-
JKLMNOPQRSTUVWXYZ
[jdoe@f33sn1 d9]$ echo {A..Z} | tr -d ' ' | cut -c 20-
TUVWXYZ

------------------------------------------------------------------------
Bonus exercise:
[jdoe@f33sn1 d9]$ python3 or python
>>> import string
>>> string.ascii_uppercase[0:5]
'ABCDE'
>>> string.ascii_uppercase[0:10]
'ABCDEFGHIJ'
>>> string.ascii_uppercase[9:]
'JKLMNOPQRSTUVWXYZ'
>>> string.ascii_uppercase[19:]
'TUVWXYZ'
>>> quit()
------------------------------------------------------------------------

#9-7 Ues "cut -b " to output in byte units.
[jdoe@f33sn1 d9]$ echo {A..Z} | tr -d ' ' | cut -b 1
A
[jdoe@f33sn1 d9]$ echo {A..Z} | tr -d ' ' | cut -b 2
B
```

```
[jdoe@f33sn1 d9]$ echo {A..Z} | tr -d ' ' | cut -b 1-10
ABCDEFGHIJ
[jdoe@f33sn1 d9]$ echo {A..Z} | tr -d ' ' | cut -b -10
ABCDEFGHIJ
[jdoe@f33sn1 d9]$ echo {A..Z} | tr -d ' ' | cut -b -20
ABCDEFGHIJKLMNOPQRST
[jdoe@f33sn1 d9]$ echo {A..Z} | tr -d ' ' | cut -b 10-
JKLMNOPQRSTUVWXYZ
[jdoe@f33sn1 d9]$ echo {A..Z} | tr -d ' ' | cut -b 20-
TUVWXYZ
```

#9-8 So, what is the main difference between the -c (character) unit and -b (bytes) unit? This is evident when you start using the East Asian Scripts (Chinese, Japanese, or Korean) in the Code.

Go to the Google translator (https://translate.google.com/?hl=en&sl=en&tl=ja&text=Hello&op=translate), and enter "Hello" in English.
On the right-hand side, you will see Konichiwa on the right-hand side. Copy it and use it on your echo command.

```
[jdoe@f33sn1 d9]$ echo "こんにちは"
こんにちは
[jdoe@f33sn1 d9]$ echo "こんにちは" | cut -b 1
▒
[jdoe@f33sn1 d9]$ echo "こんにちは" | cut -b 3
▒
[jdoe@f33sn1 d9]$ echo "こんにちは" | cut -b 5
▒
[jdoe@f33sn1 d9]$ echo "こんにちは" | cut -c 1
こ
[jdoe@f33sn1 d9]$ echo "こんにちは" | cut -c 3
に
[jdoe@f33sn1 d9]$ echo "こんにちは" | cut -c 5
は
[jdoe@f33sn1 d9]$ echo "こんにちは" | cut -c 1-
こんにちは
[jdoe@f33sn1 d9]$ echo "こんにちは" | cut -c 3-
にちは
[jdoe@f33sn1 d9]$ echo "こんにちは" | cut -c 5-
は
```

# The following exercises indicate that you will need at least 2 bytes or more bytes to express one Asian language character.

```
[jdoe@f33sn1 d9]$ echo "こんにちは" | cut -b 1-3
こ
[jdoe@f33sn1 d9]$ echo "こんにちは" | cut -b 1-4
こ▒
[jdoe@f33sn1 d9]$ echo "こんにちは" | cut -b 2-4
▒▒▒
[jdoe@f33sn1 d9]$ echo "こんにちは" | cut -b 3-6
▒ん
[jdoe@f33sn1 d9]$ echo "こんにちは" | cut -b 1-2
▒
[jdoe@f33sn1 d9]$ echo "こんにちは" | cut -b 1-3
こ
[jdoe@f33sn1 d9]$ echo "こんにちは" | cut -b 4-5
▒
[jdoe@f33sn1 d9]$ echo "こんにちは" | cut -b 4-6
ん
```

#9-9 Now, let's apply the "cut" command to our exam scores example.

```
[jdoe@f33sn1 d9]$ cat r
Peter   93      85
Diana   87      93
Tony    95      90
Lara    98      99
Bruce   80      89
[jdoe@f33sn1 d9]$ cat r | cut -f 1
Peter
Diana
Tony
Lara
Bruce
[jdoe@f33sn1 d9]$ cat r | cut -f 2
93
87
95
98
80
[jdoe@f33sn1 d9]$ cat r | cut -f 3
85
93
90
99
89
[jdoe@f33sn1 d9]$ cat r | cut -f 1,2
Peter   93
Diana   87
```

```
Tony    95
Lara    98
Bruce   80
[jdoe@f33sn1 d9]$ cat r | cut -f 2,3
93      85
87      93
95      90
98      99
80      89
```

#9-10 You cannot reverse the order of columns 2 to 1 or 3 to 1. Here, we have to use the 'awk' command.

```
[jdoe@f33sn1 d9]$ cat r | cut -f 2,1
Peter   93
Diana   87
Tony    95
Lara    98
Bruce   80
[jdoe@f33sn1 d9]$ cat r | cut -f 3,1
Peter   85
Diana   93
Tony    90
Lara    99
Bruce   89
```

#9-11 "etc/passwd" file is one of the most important files in the Linux system. Let's create a 10-line file from it and use it in our example.

```
[jdoe@f33sn1 d9]$ head /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
[jdoe@f33sn1 d9]$ head /etc/passwd > p
[jdoe@f33sn1 d9]$ nl p
     1  root:x:0:0:root:/root:/bin/bash
     2  bin:x:1:1:bin:/bin:/sbin/nologin
     3  daemon:x:2:2:daemon:/sbin:/sbin/nologin
     4  adm:x:3:4:adm:/var/adm:/sbin/nologin
     5  lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
     6  sync:x:5:0:sync:/sbin:/bin/sync
     7  shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
     8  halt:x:7:0:halt:/sbin:/sbin/halt
     9  mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
    10  operator:x:11:0:operator:/root:/sbin/nologin
```

----

*Tip
Explanation of the passwd file user properties.
E.g.) root:x:0:0:root:/root:/bin/bash

":" is a separator.

root	ID
x	password
0	UID
0	Group ID
root	Username
/root	Home directory
/bin/bash	Shell type in use

```
[jdoe@f33sn1 d9]$ cat p | cut -f 1
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
[jdoe@f33sn1 d9]$ cat p | cut -d: -f 1
root
bin
daemon
adm
lp
sync
shutdown
halt
mail
operator
[jdoe@f33sn1 d9]$ cat p | cut -d: -f 2
x
x
x
x
x
x
x
x
x
x
[jdoe@f33sn1 d9]$ cat p | cut -d: -f 3
```

```
0
1
2
3
4
5
6
7
8
11
[jdoe@f33sn1 d9]$ cat p | cut -d: -f 4
0
1
2
4
7
0
0
0
12
0
[jdoe@f33sn1 d9]$ cat p | cut -d: -f 5
root
bin
daemon
adm
lp
sync
shutdown
halt
mail
operator
[jdoe@f33sn1 d9]$ cat p | cut -d: -f 6
/root
/bin
/sbin
/var/adm
/var/spool/lpd
/sbin
/sbin
/sbin
/var/spool/mail
/root
[jdoe@f33sn1 d9]$ cat p | cut -d: -f 7
/bin/bash
/sbin/nologin
/sbin/nologin
/sbin/nologin
/sbin/nologin
/bin/sync
/sbin/shutdown
```

```
/sbin/halt
/sbin/nologin
/sbin/nologin
[jdoe@f33sn1 d9]$ cat p | cut -d: -f 1,7
root:/bin/bash
bin:/sbin/nologin
daemon:/sbin/nologin
adm:/sbin/nologin
lp:/sbin/nologin
sync:/bin/sync
shutdown:/sbin/shutdown
halt:/sbin/halt
mail:/sbin/nologin
operator:/sbin/nologin
[jdoe@f33sn1 d9]$ cat p | cut -d: -f 1,3,5
root:0:root
bin:1:bin
daemon:2:daemon
adm:3:adm
lp:4:lp
sync:5:sync
shutdown:6:shutdown
halt:7:halt
mail:8:mail
operator:11:operator
```

#9-12 When we use processes, we can use "ps -ef" command. Let's apply what we have learned to this.

```
[jdoe@f33sn1 d9]$ ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 14:52 ?        00:00:01 /usr/lib/systemd/systemd --switched-root --system --
deserializ
root         2     0  0 14:52 ?        00:00:00 [kthreadd]
root         3     2  0 14:52 ?        00:00:00 [rcu_gp]
root         4     2  0 14:52 ?        00:00:00 [rcu_par_gp]
root         6     2  0 14:52 ?        00:00:00 [kworker/0:0H-kblockd]
[...ommitted for brevity]

[jdoe@f33sn1 d9]$ ps -ef | cut -f 1
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 14:52 ?        00:00:01 /usr/lib/systemd/systemd --switched-root --system --
deserialize 30
root         2     0  0 14:52 ?        00:00:00 [kthreadd]
root         3     2  0 14:52 ?        00:00:00 [rcu_gp]
root         4     2  0 14:52 ?        00:00:00 [rcu_par_gp]
root         6     2  0 14:52 ?        00:00:00 [kworker/0:0H-kblockd]
[...ommitted for brevity]

[jdoe@f33sn1 d9]$ ps -ef | cut -f 2
UID        PID  PPID  C STIME TTY          TIME CMD
```

```
root         1       0  0 14:52 ?        00:00:01 /usr/lib/systemd/systemd --switched-root --system --
deserialize 30
root         2       0  0 14:52 ?        00:00:00 [kthreadd]
root         3       2  0 14:52 ?        00:00:00 [rcu_gp]
root         4       2  0 14:52 ?        00:00:00 [rcu_par_gp]
root         6       2  0 14:52 ?        00:00:00 [kworker/0:0H-kblockd]
[...ommitted for brevity]
[jdoe@f33sn1 d9]$ ps -ef | cut -d' ' -f 1
UID
root
root
root
root
root
[...ommitted for brevity]

#Quckly test the difference between "cut" and "awk" using who output.
jdoe     pts/0        2022-02-11 19:08 (192.168.56.1)
[jdoe@f33sn1 d9]$ who | cut -f 1
jdoe     pts/0        2022-02-11 19:08 (192.168.56.1)
[jdoe@f33sn1 d9]$ who | cut -f 2
jdoe     pts/0        2022-02-11 19:08 (192.168.56.1)
[jdoe@f33sn1 d9]$ who | cut -f 3
jdoe     pts/0        2022-02-11 19:08 (192.168.56.1)
[jdoe@f33sn1 d9]$ who | cut -d' ' -f 1
jdoe
[jdoe@f33sn1 d9]$ who | cut -d' ' -f 2

[jdoe@f33sn1 d9]$ who | cut -d' ' -f 3

[jdoe@f33sn1 d9]$ who | cut -d' ' -f 4

[jdoe@f33sn1 d9]$ who | cut -d' ' -f 5

[jdoe@f33sn1 d9]$ who | cut -d' ' -f 6
pts/0
[jdoe@f33sn1 d9]$ who | awk '{ print $1 }'
jdoe
[jdoe@f33sn1 d9]$ who | awk '{ print $2 }'
pts/0
[jdoe@f33sn1 d9]$ who | awk '{ print $3 }'
2022-02-11
[jdoe@f33sn1 d9]$ who | awk '{ print $4 }'
19:08
[jdoe@f33sn1 d9]$ who | awk '{ print $5 }'
(192.168.56.1)
[jdoe@f33sn1 d9]$ who | awk '{ print $0 }'
jdoe     pts/0        2022-02-11 19:08 (192.168.56.1)
```

Tip: 'awk' is a separate programming language and it has functions and all the features of a programming language. There are books only teaching you the use of 'awk'. Here we are only scratching the surface of Linux basics and the "awk" command.

```
# Apply the "awk" command to to "ps -ef" output.
[jdoe@f33sn1 d9]$ ps -ef | awk '{print $1 }'
UID
root
root
root
root
root
[...omitted for brevity]
[jdoe@f33sn1 d9]$ ps -ef | awk '{print $2}'
PID
1
2
3
4
6
[...omitted for brevity]
[jdoe@f33sn1 d9]$ ps -ef | awk '{print $3}'
PPID
0
0
2
2
2
[...omitted for brevity]
[jdoe@f33sn1 d9]$ ps -ef | awk '{print $0}'
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 14:52 ?        00:00:01 /usr/lib/systemd/systemd --switched-root --system --
deserialize 30
root         2     0  0 14:52 ?        00:00:00 [kthreadd]
root         3     2  0 14:52 ?        00:00:00 [rcu_gp]
root         4     2  0 14:52 ?        00:00:00 [rcu_par_gp]
root         6     2  0 14:52 ?        00:00:00 [kworker/0:0H-kblockd]
[...omitted for brevity]
[jdoe@f33sn1 d9]$ ps -ef | awk '{print $1, $3}'
UID PPID
root 0
root 0
root 2
root 2
root 2
[...omitted for brevity]
[jdoe@f33sn1 d9]$ ps -ef | awk '{print $1, $3, $5}'
UID PPID STIME
root 0 14:52
root 0 14:52
root 2 14:52
root 2 14:52
root 2 14:52
[...omitted for brevity]
[jdoe@f33sn1 d9]$ ps -ef | awk '{print $1, "\t",  $3, "\t", $5}'
UID     PPID    STIME
root    0       14:52
root    0       14:52
root    2       14:52
root    2       14:52
root    2       14:52
[...omitted for brevity]
```

```
[jdoe@f33sn1 ex09]$ cat p | awk -F: '{print $1,$7}'
root /bin/bash
bin /sbin/nologin
daemon /sbin/nologin
adm /sbin/nologin
lp /sbin/nologin
sync /bin/sync
shutdown /sbin/shutdown
halt /sbin/halt
mail /sbin/nologin
operator /sbin/nologin

[jdoe@f33sn1 ex09]$ cat p | awk -F: '{print $1, "\t",  $3, "\t", $5}'
root     0       root
bin      1       bin
daemon   2       daemon
adm      3       adm
lp       4       lp
sync     5       sync
shutdown         6       shutdown
halt     7       halt
mail     8       mail
operator         11      operator
[jdoe@f33sn1 ex09]$ cat p | awk -F: '{print $1, "\t",  $3, "\t", $7}'
root     0       /bin/bash
bin      1       /sbin/nologin
daemon   2       /sbin/nologin
adm      3       /sbin/nologin
lp       4       /sbin/nologin
sync     5       /bin/sync
shutdown         6       /sbin/shutdown
halt     7       /sbin/halt
mail     8       /sbin/nologin
operator         11      /sbin/nologin
```

#9-13 Now use the "awk" command to the student score file r.

```
[jdoe@f33sn1 d9]$ cat r | awk '{print $0}'
```

```
Peter   93      85
Diana   87      93
Tony    95      90
Lara    98      99
Bruce   80      89
[jdoe@f33sn1 d9]$ cat r | awk '{print $1}'
Peter
Diana
Tony
Lara
Bruce
[jdoe@f33sn1 d9]$ cat r | awk '{print $2}'
93
87
95
98
80
[jdoe@f33sn1 d9]$ cat r | awk '{print $3}'
85
93
90
99
89

# Reverse the columns as you wish.
[jdoe@f33sn1 d9]$ cat r | awk '{print $3, $1, $2}'
85 Peter 93
93 Diana 87
90 Tony 95
99 Lara 98
89 Bruce 80

# Add extra descriptions for each column item.
[jdoe@f33sn1 d9]$ cat r | awk '{print "Student", $1, "English : ", $2, "Math : ", $3}'
Student Peter English :  93 Math :  85
Student Diana English :  87 Math :  93
Student Tony English :  95 Math :  90
Student Lara English :  98 Math :  99
Student Bruce English :  80 Math :  89
[jdoe@f33sn1 d9]$ cat r | awk '{print "Student", $1, ", English : ", $2, ", Math : ", $3}'
Student Peter , English :  93 , Math :  85
Student Diana , English :  87 , Math :  93
Student Tony , English :  95 , Math :  90
Student Lara , English :  98 , Math :  99
Student Bruce , English :  80 , Math :  89
[jdoe@f33sn1 d9]$ cat r | awk '{print "English : ", $2, ", Math : ", $3, "Name : ", $1}'
English :  93 , Math :  85 Name :  Peter
English :  87 , Math :  93 Name :  Diana
English :  95 , Math :  90 Name :  Tony
English :  98 , Math :  99 Name :  Lara
English :  80 , Math :  89 Name :  Bruce
```

```
# You can use "\b" to negate the spaces and add strings together.
[jdoe@f33sn1 d9]$ cat r | awk '{print $1, "\b_classA" ", English : ", $2, ", Math : ", $3}'
Peter_classA, English :  93 , Math :  85
Diana_classA, English :  87 , Math :  93
Tony_classA, English :  95 , Math :  90
Lara_classA, English :  98 , Math :  99
Bruce_classA, English :  80 , Math :  89
```

===

Ex9 Summary

Use 'man cut' to study the options.

```
[jdoe@f33sn1 ~]$ man cut
```

```
NAME:
    cut - remove sections from each line of files

SYNOPSIS:
    cut OPTION... [FILE]...

OPTIONS:
    -b, --bytes=LIST
        select only these bytes

    -c, --characters=LIST
        select only these characters

    -d, --delimiter=DELIM
        use DELIM instead of TAB for the field delimiter.

    -f, --fields=LIST
        select  only  these fields;  also print any line that contains no delimiter character, unless the
        -s option is specified

Range selection options:
    N     N'th byte, character, or field, counted from 1

    N-    from N'th byte, character or field, to end of line

    N-M   from N'th to M'th (included) byte, character, or field

    -M    from first to M'th (included) byte, character, or field
```

===

Exercise 10: sort

time: 15 minutes
==

A quick explanation of "man sort"

```
[jdoe@f33sn1 ~]$ man sort
```

OPTIONS:
-b: ignore leading space
-f: ignore casing
-n: sort by number
-R: Random sort using has value
-r: reverse sort
-h: user (human) friendly (2K, 1G)

Extended OPTIONS:
-c: Check sort
-k n: sort by nth field
-m: Merger already sorted files
-o: Output to a file.
-t: Specify deliminator, default deleminator is a space.
-u: Sort and remove duplicates, unique.

#10-1

```
[jdoe@f33sn1 ~]$ mkdir d10
[jdoe@f33sn1 ~]$ cp d8/* d10
[jdoe@f33sn1 ~]$ cd d10
[jdoe@f33sn1 d10]$ ls
e1  eng  m1  math  name  result1  result2  result3
[jdoe@f33sn1 d10]$ rm e1 m1 result*
[jdoe@f33sn1 d10]$ ls
eng  math  name
[jdoe@f33sn1 d10]$ sort eng | nl
     1  80
     2  87
     3  93
     4  95
     5  98
[jdoe@f33sn1 d10]$ cat > eng       << Change the values
99
85
99
85
90
[jdoe@f33sn1 d10]$ cat eng
99
85
99
85
```

```
90

# Default sorting method is ascending.
[jdoe@f33sn1 d10]$ sort eng
85
85
90
99
99

# User "-r" option with "sort" command to reverse.
[jdoe@f33sn1 d10]$ sort eng -r | nl
     1  99
     2  99
     3  90
     4  85
     5  85

# Add "-u" option for reverse.
[jdoe@f33sn1 d10]$ sort eng -r -u | nl
     1  99
     2  90
     3  85

#10-2 Sort two files.
[jdoe@f33sn1 d10]$ sort eng math
85
85
85
89
90
90
93
99
99
99

[jdoe@f33sn1 d10]$ sort eng math | nl
     1  85
     2  85
     3  85
     4  89
     5  90
     6  90
     7  93
     8  99
     9  99
    10  99

[jdoe@f33sn1 d10]$ sort eng math -u | nl
     1  85
     2  89
     3  90
     4  93
     5  99

#10-3 Use the "-m" option to merge files. This just merges values.
[jdoe@f33sn1 d10]$ sort eng math -m
85
93
90
99
85
99
85
90
99
89

[jdoe@f33sn1 d10]$ sort eng math -m | nl
     1  100
     2  85
     3  90
     4  100
     5  80
     6  90
     7  93
     8  90
     9  99
    10  89


#10-4 Use option "-o" to print the output.
[jdoe@f33sn1 d10]$ sort eng math -o em1
[jdoe@f33sn1 d10]$ nl em1
     1  85
     2  85
     3  85
     4  89
     5  90
     6  90
     7  93
     8  99
     9  99
    10  99

[jdoe@f33sn1 d10]$ sort eng math -m -o em2
[jdoe@f33sn1 d10]$ nl em2
     1  85
     2  93
     3  90
     4  99
     5  85
     6  99
     7  85
     8  90
     9  99
    10  89
```

#10-5 Sorting names using alphabet order.

```
[jdoe@f33sn1 d10]$ nl name
     1  Peter
     2  Diana
     3  Tony
     4  Lara
     5  Bruce
[jdoe@f33sn1 d10]$ sort name
Bruce
Diana
Lara
Peter
Tony
[jdoe@f33sn1 d10]$ sort name | nl
     1  Bruce
     2  Diana
     3  Lara
     4  Peter
     5  Tony
[jdoe@f33sn1 d10]$ sort name -r | nl
     1  Tony
     2  Peter
     3  Lara
     4  Diana
     5  Bruce
```

#10-6 Use 'paste' and 'tee' commands to merge three files into one and also display the result on the screen.

```
[jdoe@f33sn1 d10]$ paste name eng math | tee nem
Peter   99      85
Diana   85      93
Tony    99      90
Lara    85      99
Bruce   90      89
[jdoe@f33sn1 d10]$ nl nem
     1  Peter   99      85
     2  Diana   85      93
     3  Tony    99      90
     4  Lara    85      99
     5  Bruce   90      89
[jdoe@f33sn1 d10]$ sort nem | nl
     1  Bruce   90      89
     2  Diana   85      93
     3  Lara    85      99
     4  Peter   99      85
     5  Tony    99      90

# Use the "-k" option to sore using field 2 as the primary key.
[jdoe@f33sn1 d10]$ sort -k 2 nem | nl
     1  Diana   85      93
     2  Lara    85      99
     3  Bruce   90      89
     4  Peter   99      85
     5  Tony    99      90
[jdoe@f33sn1 d10]$ sort -k 2 -r  nem | nl
     1  Tony    99      90
     2  Peter   99      85
     3  Bruce   90      89
     4  Lara    85      99
     5  Diana   85      93

# Use field 3 for sorting.
[jdoe@f33sn1 d10]$ sort -k 3 nem | nl
     1  Peter   99      85
     2  Bruce   90      89
     3  Tony    99      90
     4  Diana   85      93
     5  Lara    85      99
[jdoe@f33sn1 d10]$ sort -k 3 -r nem | nl
     1  Lara    85      99
     2  Diana   85      93
     3  Tony    99      90
     4  Bruce   90      89
     5  Peter   99      85
[jdoe@f33sn1 d10]$ sort -k 1 -r  nem | nl
     1  Tony    99      90
     2  Peter   99      85
     3  Lara    85      99
     4  Diana   85      93
     5  Bruce   90      89

#10-7 Use the "-R" option to shuffle the items, at random.
[jdoe@f33sn1 d10]$ sort eng -R | nl
     1  85
     2  85
     3  99
     4  99
     5  90
[jdoe@f33sn1 d10]$ sort eng -R | nl
     1  90
     2  99
     3  99
     4  85
     5  85
[jdoe@f33sn1 d10]$ sort eng -R | nl
```

```
     1  99
     2  99
     3  85
     4  85
     5  90

#10-8 Check if the items are sorted already.
[jdoe@f33sn1 d10]$ nl eng
     1  99
     2  85
     3  99
     4  85
     5  90
[jdoe@f33sn1 d10]$ sort -c eng
sort: eng:2: disorder: 85
[jdoe@f33sn1 d10]$ echo $?           <<< "echo $?" allows you to check the status of the previous
command. 1=error, 0=OK.
1

# Sort the values and save them to a new file called eng1.
[jdoe@f33sn1 d10]$ sort eng > eng1
[jdoe@f33sn1 d10]$ nl eng1
     1  85
     2  85
     3  90
     4  99
     5  99
[jdoe@f33sn1 d10]$ sort -c eng1       <<< Silent YES. In Linux, if there is no output, all is GOOD!
[jdoe@f33sn1 d10]$ echo $?            <<< "echo $?" allows you to check the status of the previous
command. 1=error, 0=OK.
0

#10-9 Sort exercise using "/etc/passwd".

[jdoe@f33sn1 d10]$ head -12 /etc/passwd > p
[jdoe@f33sn1 d10]$ nl p
     1  root:x:0:0:root:/root:/bin/bash
     2  bin:x:1:1:bin:/bin:/sbin/nologin
     3  daemon:x:2:2:daemon:/sbin:/sbin/nologin
     4  adm:x:3:4:adm:/var/adm:/sbin/nologin
     5  lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
     6  sync:x:5:0:sync:/sbin:/bin/sync
     7  shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
     8  halt:x:7:0:halt:/sbin:/sbin/halt
     9  mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
    10  operator:x:11:0:operator:/root:/sbin/nologin
[jdoe@f33sn1 d10]$ sort p | nl
     1  adm:x:3:4:adm:/var/adm:/sbin/nologin
     2  bin:x:1:1:bin:/bin:/sbin/nologin
     3  daemon:x:2:2:daemon:/sbin:/sbin/nologin
     4  halt:x:7:0:halt:/sbin:/sbin/halt
     5  lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
     6  mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
     7  operator:x:11:0:operator:/root:/sbin/nologin
     8  root:x:0:0:root:/root:/bin/bash
     9  shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
    10  sync:x:5:0:sync:/sbin:/bin/sync

# Sort by using the third field. The first output is unchanged, you have to use "-t:" to specify the
separator. n the second output, now it is sorted using the third field.
[jdoe@f33sn1 d10]$ sort -k 3 p | nl
     1  adm:x:3:4:adm:/var/adm:/sbin/nologin
     2  bin:x:1:1:bin:/bin:/sbin/nologin
     3  daemon:x:2:2:daemon:/sbin:/sbin/nologin
     4  halt:x:7:0:halt:/sbin:/sbin/halt
     5  lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
     6  mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
     7  operator:x:11:0:operator:/root:/sbin/nologin
     8  root:x:0:0:root:/root:/bin/bash
     9  shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
    10  sync:x:5:0:sync:/sbin:/bin/sync
[jdoe@f33sn1 d10]$ sort -t: -k  3 p | nl            <<<Sorted using third field.
     1  root:x:0:0:root:/root:/bin/bash
     2  operator:x:11:0:operator:/root:/sbin/nologin     <<< still have an issue, this is number 11. Have
to fix this issue.
     3  bin:x:1:1:bin:/bin:/sbin/nologin
     4  daemon:x:2:2:daemon:/sbin:/sbin/nologin
     5  adm:x:3:4:adm:/var/adm:/sbin/nologin
     6  lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
     7  sync:x:5:0:sync:/sbin:/bin/sync
     8  shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
     9  halt:x:7:0:halt:/sbin:/sbin/halt
    10  mail:x:8:12:mail:/var/spool/mail:/sbin/nologin

# Use the "-n" option to fix the above issue.
[jdoe@f33sn1 d10]$ sort -t: -k  3 -n p | nl
     1  root:x:0:0:root:/root:/bin/bash
     2  bin:x:1:1:bin:/bin:/sbin/nologin
     3  daemon:x:2:2:daemon:/sbin:/sbin/nologin
     4  adm:x:3:4:adm:/var/adm:/sbin/nologin
     5  lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
     6  sync:x:5:0:sync:/sbin:/bin/sync
     7  shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
     8  halt:x:7:0:halt:/sbin:/sbin/halt
     9  mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
    10  operator:x:11:0:operator:/root:/sbin/nologin

# If we apply the awk command and get the specific fields only. You can do this.
[jdoe@f33sn1 d10]$ sort -t: -k 3 p | awk -F: '{print $1, $3}' | nl
     1  root 0
     2  operator 11
     3  bin 1
     4  daemon 2
     5  adm 3
     6  lp 4
     7  sync 5
     8  shutdown 6
     9  halt 7
    10  mail 8
[jdoe@f33sn1 d10]$ sort -t: -k 3 -n p | awk -F: '{print $1, $3}' | nl
     1  root 0
     2  bin 1
     3  daemon 2
     4  adm 3
     5  lp 4
     6  sync 5
     7  shutdown 6
     8  halt 7
     9  mail 8
    10  operator 11
[jdoe@f33sn1 d10]$ sort -t: -k 3 -n p | awk -F: '{print $1, $2, $3}' | nl
     1  root x 0
     2  bin x 1
     3  daemon x 2
     4  adm x 3
     5  lp x 4
     6  sync x 5
     7  shutdown x 6
     8  halt x 7
     9  mail x 8
    10  operator x 11

#10-10
[jdoe@f33sn1 d10]$ ll /var/log/
total 9096
drwxr-xr-x. 2 root   root             4096 Mar 19  2021 anaconda
drwx------. 2 root   root               23 Mar 19  2021 audit
-rw-------. 1 root   root             3296 Feb 13 12:34 boot.log
-rw-------. 1 root   root            16363 Mar 26  2021 boot.log-20210326
-rw-------. 1 root   root            16511 Feb  4 17:01 boot.log-20220204
-rw-------. 1 root   root            17113 Feb  7 11:11 boot.log-20220207
-rw-------. 1 root   root            32435 Feb  8 09:49 boot.log-20220208
-rw-------. 1 root   root             3185 Feb  9 00:00 boot.log-20220209
-rw-------. 1 root   root            12945 Feb 11 14:52 boot.log-20220211
-rw-------. 1 root   root            16697 Feb 13 12:34 boot.log-20220213
-rw-rw----. 1 root   utmp             2304 Feb 11 15:25 btmp
-rw-rw----. 1 root   utmp              384 Mar 19  2021 btmp-20220204
drwxr-xr-x. 2 chrony chrony              6 Sep 16  2020 chrony
[...omitted for brevity]

[jdoe@f33sn1 d10]$ ll /var/log/ | sort -k 5
total 9096
-rw-------. 1 root   root                0 Feb 11 14:52 maillog-20220213
```

```
-rw-------. 1 root   root              0 Feb 11 14:52 spooler-20220213
-rw-------. 1 root   root              0 Feb 13 12:34 maillog
-rw-------. 1 root   root              0 Feb 13 12:34 spooler
-rw-------. 1 root   root              0 Feb  4 17:01 maillog-20220211
-rw-------. 1 root   root              0 Feb  4 17:01 spooler-20220211
-rw-r-----. 1 root   root              0 Mar 19  2021 firewalld
-rw-------. 1 root   root              0 Mar 19  2021 maillog-20220204
-rw-------. 1 root   root              0 Mar 19  2021 spooler-20220204
-rw-------. 1 root   root              0 Mar 19  2021 tallylog
drwxrwx---. 2 apache root            101 Feb 13 12:34 php-fpm
-rw-------. 1 root   root            102 Feb 11 18:20 hawkey.log-20220213
-rw-r--r--. 1 root   root           1040 Sep 20  2020 README
-rw-------. 1 root   root          10744 Feb 13 12:50 dnf.librepo.log
-rw-------. 1 root   root         121003 Feb  4 17:01 secure-20220204
-rw-------. 1 root   root         128198 Feb 11 14:52 secure-20220211
-rw-------. 1 root   root          12945 Feb 11 14:52 boot.log-20220211       <<< Problem, sorted by character
-rw-------. 1 root   root           1445 Feb 13 12:34 secure-20220213         <<< Problem, sorted by character
drwx------. 2 root   root            146 Feb 13 12:34 httpd                   <<< Problem, sorted by character
-rw-------. 1 root   root          16363 Mar 26  2021 boot.log-20210326
-rw-------. 1 root   root          16511 Feb  4 17:01 boot.log-20220204
[...ommitted for brevity]
[jdoe@f33sn1 d10]$ ll /var/log/ | sort -k 5 -n
-rw-------. 1 root   root              0 Feb 11 14:52 maillog-20220213
-rw-------. 1 root   root              0 Feb 11 14:52 spooler-20220213
-rw-------. 1 root   root              0 Feb 13 12:34 maillog
-rw-------. 1 root   root              0 Feb 13 12:34 spooler
-rw-------. 1 root   root              0 Feb  4 17:01 maillog-20220211
-rw-------. 1 root   root              0 Feb  4 17:01 spooler-20220211
-rw-------. 1 root   root              0 Mar 19  2021 maillog-20220204
-rw-------. 1 root   root              0 Mar 19  2021 spooler-20220204
-rw-------. 1 root   root              0 Mar 19  2021 tallylog
-rw-r-----. 1 root   root              0 Mar 19  2021 firewalld
total 9096
drwx------. 2 root   root              6 Mar 19  2021 private
drwxr-xr-x. 2 chrony chrony            6 Sep 16  2020 chrony
drwx------. 3 root   root             17 Mar 19  2021 samba
drwx------. 2 root   root             23 Mar 19  2021 audit
drwxr-sr-x+ 3 root   systemd-journal  46 Mar 19  2021 journal
-rw-------. 1 root   root             51 Feb 13 12:44 hawkey.log
[...ommitted for brevity]
[jdoe@f33sn1 d10]$ ll /var/log/ | sort -k 5 -n -r
-rw-------. 1 root   root        4265170 Feb 11 14:52 messages-20220211
-rw-------. 1 root   root        3603426 Feb  4 17:01 messages-20220204
-rw-rw-r--. 1 root   utmp         292876 Feb 13 12:34 lastlog
-rw-r--r--. 1 root   root         292760 Feb 13 12:50 dnf.log
-rw-------. 1 root   root         217340 Feb 13 12:34 messages-20220213
-rw-------. 1 root   root         178336 Feb 10 05:32 dnf.librepo.log-20220211
-rw-rw-r--. 1 root   utmp         175872 Feb 13 12:34 wtmp
-rw-------. 1 root   root         128198 Feb 11 14:52 secure-20220211
```

```
-rw-------. 1 root   root          121003 Feb  4 17:01 secure-20220204
-rw-------. 1 root   root           32435 Feb  8 09:49 boot.log-20220208
-rw-r--r--. 1 root   root           30826 Mar 23  2021 dnf.librepo.log-20220204
-rw-------. 1 root   root           23384 Feb 11 21:11 dnf.librepo.log-20220213
-rw-------. 1 root   root           22073 Feb 13 13:34 messages
-rw-------. 1 root   root           17113 Feb  7 11:11 boot.log-20220207
-rw-------. 1 root   root           16697 Feb 13 12:34 boot.log-20220213
[...omitted for brevity]

#10-11 "df" = disk free, used to check partition size and usage.
[jdoe@f33sn1 d10]$ df
Filesystem                 1K-blocks    Used Available Use% Mounted on
devtmpfs                      476448       0    476448   0% /dev
tmpfs                         498392       0    498392   0% /dev/shm
tmpfs                         199360     992    198368   1% /run
/dev/mapper/fedora_fedora-root   7325696 2159340   5166356  30% /
tmpfs                         498392       4    498388   1% /tmp
/dev/sda1                    1038336  203340    834996  20% /boot
tmpfs                          99676       0     99676   0% /run/user/1000
[jdoe@f33sn1 d10]$ df | sort -k 5
devtmpfs                      476448       0    476448   0% /dev
tmpfs                         498392       0    498392   0% /dev/shm
tmpfs                          99676       0     99676   0% /run/user/1000
tmpfs                         199360     992    198368   1% /run
tmpfs                         498392       4    498388   1% /tmp
/dev/sda1                    1038336  203340    834996  20% /boot
/dev/mapper/fedora_fedora-root   7325696 2159340   5166356  30% /
Filesystem                 1K-blocks    Used Available Use% Mounted on
[jdoe@f33sn1 d10]$ df | sort -k 2 -n
Filesystem                 1K-blocks    Used Available Use% Mounted on
tmpfs                          99676       0     99676   0% /run/user/1000
tmpfs                         199360     992    198368   1% /run
devtmpfs                      476448       0    476448   0% /dev
tmpfs                         498392       0    498392   0% /dev/shm
tmpfs                         498392       4    498388   1% /tmp
/dev/sda1                    1038336  203340    834996  20% /boot
/dev/mapper/fedora_fedora-root   7325696 2159340   5166356  30% /
[jdoe@f33sn1 d10]$ df | sort -k 2 -n -h                    <<< Option "-n" and "-h" are incompatible.
sort: options '-hn' are incompatible
[jdoe@f33sn1 d10]$ df | sort -k 2 -h                    <<< Option "-h"
tmpfs                          99676       0     99676   0% /run/user/1000
tmpfs                         199360     992    198368   1% /run
devtmpfs                      476448       0    476448   0% /dev
tmpfs                         498392       0    498392   0% /dev/shm
tmpfs                         498392       4    498388   1% /tmp
/dev/sda1                    1038336  203340    834996  20% /boot
/dev/mapper/fedora_fedora-root   7325696 2159340   5166356  30% /
Filesystem                 1K-blocks    Used Available Use% Mounted on
```

==

Ex10 Summary

Review "man sort"

[jdoe@f33sn1 ~]$ man sort

==

Exercise 11: uniq

Time: 15 minutes

```
==============================================================================

uniq - report or omit repeated lines

Usage:
    uniq [OPTION]... [INPUT [OUTPUT]]

        High usage options:
    -c, --count
        prefix lines by the number of occurrences
    -d, --repeated
        only print duplicate lines, one for each group
    -D   print all duplicate lines
    -u, --unique
        only print unique lines
    -i, --ignore-case
        ignore differences in case when comparing

        Low usage options:
    -w, --check-chars=N
        compare no more than N characters in lines
    -s, --skip-chars=N
        avoid comparing the first N characters
    -f, --skip-fields=N
        avoid comparing the first N fields


shuf - generate random permutations

Usage:
    shuf [OPTION]... [FILE]
    shuf -e [OPTION]... [ARG]...
    shuf -i LO-HI [OPTION]...

        High usage options:
    -e, --echo
                treat each ARG as an input line
    -i, --input-range=LO-HI
        treat each number LO through HI as an input line
    -n, --head-count=COUNT
        output at most COUNT lines
    -o, --output=FILE
        write result to FILE instead of standard output

        Low usage options:
    -r, --repeat
        output lines can be repeated
==============================================================================
```

#uniq command exercises

Ex11-1 Make a directory ex11 and change the working directory. Then create a file b as shown.

```
bchoi@pop-os:~$ mkdir ex11 && cd ex11
bchoi@pop-os:~/ex11$ cat > b
one
two
two
three
three
three
one
two
two
bchoi@pop-os:~/ex11$ nl b
     1	one
     2	two
     3	two
     4	three
     5	three
     6	three
     7	one
     8	two
     9	two
```

Ex11-2 Use the uniq command with nl to see the result of the command. You will see that uniq only find the uniqe values for the continuous occurrences.

```
bchoi@pop-os:~/ex11$ uniq b | nl
     1	one
     2	two
     3	three
     4	one
     5	two
```

Ex11-3 Now to find the unique values, run the combination of the "sort" and 'uniq" commands.

```
bchoi@pop-os:~/ex11$ sort b | nl
     1	one
     2	one
     3	three
     4	three
     5	three
     6	two
     7	two
     8	two
     9	two
bchoi@pop-os:~/ex11$ sort b | uniq
one
three
two
```

Ex11-4 Count the number of occurrences using the "-c" or "--count" option.

```
choi@pop-os:~/ex11$ uniq b | nl
     1	one
     2	two
     3	three
     4	one
     5	two
bchoi@pop-os:~/ex11$ uniq -c b | nl
     1	      1 one
     2	      2 two
     3	      3 three
     4	      1 one
     5	      2 two
bchoi@pop-os:~/ex11$ nl b
     1	one
     2	two
     3	two
     4	three
     5	three
     6	three
     7	one
     8	two
     9	two
```

Ex11-5 Output non-repeated single line using the "-u" option.

```
bchoi@pop-os:~/ex11$ uniq -u b | nl
     1	one
     2	one
bchoi@pop-os:~/ex11$ nl b
     1	one	<<< non-repeated single line
     2	two
     3	two
     4	three
     5	three
     6	three
     7	one	<<< non-repeated single line
     8	two
     9	two
```

Ex11-6 Output repeated line using the "-d" option.

```
bchoi@pop-os:~/ex11$ uniq -d b | nl
     1	two
     2	three
     3	two
bchoi@pop-os:~/ex11$ nl b
     1	one
     2	two
     3	two	<<< repeated line
     4	three
     5	three	<<< repeated line
     6	three	<<< repeated line
     7	one
     8      two
     9      two      <<< repeated line
```

```
# Add the "-c" option to count the number of occurrences.
bchoi@pop-os:~/ex11$ uniq -d -c b | nl
     1          2 two
     2          3 three
     3          2 two
```

Ex11-7 Only display repeated lines using the "-D" option. This option has the opposite effect of the "-u" option.

```
bchoi@pop-os:~/ex11$ uniq -D b | nl
     1      two
     2      two
     3      three
     4      three
     5      three
     6      two
     7      two
bchoi@pop-os:~/ex11$ uniq -u b | nl
     1      one
     2      one
```

Ex11-8 Treat the repeated lines as a single line using the "-i" option.

```
bchoi@pop-os:~/ex11$ uniq -i b | nl
     1      one
     2      two
     3      three
     4      one
     5      two
```

Ex11-9 Creat file k as shown below.

```
bchoi@pop-os:~/ex11$ cat > k
Peter   85      90
Diana   90      90
Tony    90      90
Lara    80      80
Bruce   80      80
bchoi@pop-os:~/ex11$ nl k
     1      Peter   85      90
     2      Diana   90      90
     3      Tony    90      90
     4      Lara    80      80
     5      Bruce   80      80
```

Ex11-10 Use the "-f" option to filter the repeated lines by field number.

```
bchoi@pop-os:~/ex11$ uniq k
Peter   85      90
Diana   90      90
Tony    90      90
Lara    80      80
```

```
Bruce   80      80
bchoi@pop-os:~/ex11$ uniq k -f 0
Peter   85      90
Diana   90      90
Tony    90      90
Lara    80      80
Bruce   80      80
bchoi@pop-os:~/ex11$ uniq k -f 1
Peter   85      90
Diana   90      90
Lara    80      80
bchoi@pop-os:~/ex11$ uniq k -f 2
Peter   85      90
Lara    80      80
```

#shuf command exercises

Ex11-11 Use the "cat > eng" command to create the following file. Run the 'shuf eng" command a few times to observe the shuffling numbers.

```
bchoi@pop-os:~/ex11$ cat > eng
85
90
90
80
80
bchoi@pop-os:~/ex11$ nl eng
     1	85
     2	90
     3	90
     4	80
     5	80
bchoi@pop-os:~/ex11$ shuf eng
90
85
80
90
80
bchoi@pop-os:~/ex11$ shuf eng
80
80
85
90
90
```

Ex11.12 Now use the "-e" option to shuffle a string of numbers.

```
bchoi@pop-os:~/ex11$ cat > eng
85
90
90
```

```
80
80
bchoi@pop-os:~/ex11$ nl eng
     1      85
     2      90
     3      90
     4      80
     5      80
bchoi@pop-os:~/ex11$ shuf eng
90
85
80
90
80
bchoi@pop-os:~/ex11$ shuf eng
80
80
85
90
90
```

Ex11.13 Now use your favorite cities to shuffle.

```
bchoi@pop-os:~/ex11$ shuf -e Sydney LA London Singapore Tokyo
LA
Sydney
Singapore
Tokyo
London
bchoi@pop-os:~/ex11$ shuf -e Sydney LA London Singapore Tokyo
Singapore
LA
Tokyo
Sydney
London
bchoi@pop-os:~/ex11$ shuf -e Sydney LA London Singapore Tokyo
Tokyo
LA
London
Singapore
Sydney
```

Ex11.14 Add "-n 1" option to call out a random city, and use "-n 3" to call three random cities from the list.

```
bchoi@pop-os:~/ex11$ shuf -e Sydney LA London Singapore Tokyo -n 1
London
bchoi@pop-os:~/ex11$ shuf -e Sydney LA London Singapore Tokyo -n 1
Singapore
bchoi@pop-os:~/ex11$ shuf -e Sydney LA London Singapore Tokyo -n 1
LA
bchoi@pop-os:~/ex11$ shuf -e Sydney LA London Singapore Tokyo -n 3
London
```

```
Singapore
Tokyo
bchoi@pop-os:~/ex11$ shuf -e Sydney LA London Singapore Tokyo -n 3
LA
Sydney
London
bchoi@pop-os:~/ex11$ shuf -e Sydney LA London Singapore Tokyo -n 3
London
Tokyo
Singapore
```

Ex11.15 Who is going to the shop pick up our lunch?

```
bchoi@pop-os:~/ex11$ shuf -e John George Erwin Shannon Colin -n 1
Erwin
bchoi@pop-os:~/ex11$ shuf -e John George Erwin Shannon Colin -n 1
Colin
bchoi@pop-os:~/ex11$ shuf -e John George Erwin Shannon Colin -n 1
George
```

Ex11.16 Practice what we have learnt so far.

```
bchoi@pop-os:~/ex11$ shuf eng | nl
     1	90
     2	85
     3	80
     4	80
     5	90
bchoi@pop-os:~/ex11$ shuf eng -n 1 | nl
     1	90
bchoi@pop-os:~/ex11$ shuf eng -n 2 | nl
     1	90
     2	90
bchoi@pop-os:~/ex11$ shuf eng -n 3 | nl
     1	90
     2	80
     3	85
```

Ex11.17 Practice shuf using a number range of 1-10.

```
bchoi@pop-os:~/ex11$ shuf eng -n 1 | nl
     1	90
bchoi@pop-os:~/ex11$ shuf eng -n 2 | nl
     1	90
     2	90
bchoi@pop-os:~/ex11$ shuf eng -n 3 | nl
     1	90
     2	80
     3	85
```

Ex11.18 Using "shuf -i" command, let's get six random numbers to fill out the Saturday night Lotto.

```
bchoi@pop-os:~/ex11$ shuf -i 1-45 -n 6
37
36
```

```
6
22
23
16
bchoi@pop-os:~/ex11$ shuf -i 1-45 -n 6
24
6
29
19
3
43
bchoi@pop-os:~/ex11$ shuf -i 1-45 -n 6
12
40
24
21
39
30
```

Ex11.19 Prettify the output per column by adding the "paste" option.

```
bchoi@pop-os:~/ex11$ shuf -i 1-45 -n 6 | paste - -
32      35
23      28
22      8
bchoi@pop-os:~/ex11$ shuf -i 1-45 -n 6 | paste - - -
29      30      44
9       36      23
bchoi@pop-os:~/ex11$ shuf -i 1-45 -n 6 | paste - - -
6       22      9
28      26      37
```

Tip - Getting the six random numbers using Python.

```
bchoi@pop-os:~/ex11$ python3
Python 3.9.7 (default, Sep 10 2021, 14:59:43)
[GCC 11.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import random
>>> list = random.sample(range(1, 46), 6)
>>> print(list)
[32, 30, 29, 22, 10, 40]
>>> def lotto():
...     list = random.sample(range(1, 46), 6)
...     print(list)
...
>>> lotto()
[32, 8, 21, 19, 44, 20]
>>> lotto()
[38, 16, 19, 8, 35, 21]
>>> lotto()
[6, 26, 2, 7, 13, 37]
```

Tip2 - Save the application as a python file and run it to get your lucky numbers.

```
bchoi@pop-os:~/ex11$ cat > luckynum.py
import random
luckynums = random.sample(range(1, 46), 6)
print(luckynums)
bchoi@pop-os:~/ex11$ cat luckynum.py
import random
luckynums = random.sample(range(1, 46), 6)
print(luckynums)
bchoi@pop-os:~/ex11$ python3 luckynum.py
[5, 39, 3, 37, 1, 42]
bchoi@pop-os:~/ex11$ python3 luckynum.py
[12, 18, 2, 3, 10, 30]
bchoi@pop-os:~/ex11$ python3 luckynum.py
[24, 32, 12, 2, 8, 42]
```

Ex11.20 Use the "-r" option to repeat the shuf command.

```
bchoi@pop-os:~/ex11$ shuf -i 1-10
2
7
8
10
4
9
6
3
1
5
bchoi@pop-os:~/ex11$ shuf -i 1-10 -r
2
4
7
6
7
4
10
6
1
6
4
10
2
6
4
7
8
6
4
[...omitted for brevity]
```

EX11.21 Saving shuf output to a file.

```
bchoi@pop-os:~/ex11$ shuf -i 1-45 -n 6 -o getlucky1
bchoi@pop-os:~/ex11$ more getlucky1
35
28
38
14
23
26
more: cannot open getlucky3: No such file or directory
bchoi@pop-os:~/ex11$
bchoi@pop-os:~/ex11$ more getlucky2
30
42
17
9
19
29
```

===

Ex11 Summary

# Getting more help on the command.

```
bchoi@pop-os:~/ex11$ uniq --help
bchoi@pop-os:~/ex11$ man uniq

bchoi@pop-os:~/ex11$ shuf --help
bchoi@pop-os:~/ex11$ man shuf
```

Use the man page to go over a quick summary.

===

Exercise 12: tr (trim)

tr - cut, change, trim text

18 mins
===

tr, expand, unexpand --20

tr - translate or delete characters

```
SYNOPSIS
    tr [OPTION]... SET1 [SET2]

Usage:
    -d, --delete
        delete characters in SET1, do not translate

    -s, --squeeze-repeats
        replace each sequence of a repeated character that is
    -t truncate
        change characters
    -c, -C, --complement
        use the complement of SET1
```

expand - convert tabs to spaces

unexpand - convert spaces to tabs

Ex12-1 Make a new directory and change the working directory.

```
bchoi@pop-os:~$ mkdir ex12 && cd ex12
bchoi@pop-os:~/ex12$ echo "cis"
cis
bchoi@pop-os:~/ex12$ echo "cis" | tr -d "c"
is
bchoi@pop-os:~/ex12$ echo "cis" | tr -d "c" | nl
     1	is
bchoi@pop-os:~/ex12$ echo "ciscis" | tr -d "c" | nl
     1	isis
bchoi@pop-os:~/ex12$ echo "ccccciiiiisssssscis" | tr -d "c" | nl
     1	iiiiissssssis
bchoi@pop-os:~/ex12$ echo "ccccciiiiisssssscis" | tr -d "i" | nl
     1	cccccsssssscs
bchoi@pop-os:~/ex12$ echo "ccccciiiiisssssscis" | tr -d "s" | nl
     1	ccccciiiiici
bchoi@pop-os:~/ex12$ echo "ccccciiiiisssssscis" | tr -d "cis" | nl

bchoi@pop-os:~/ex12$ echo "ccccciiiiisssssscis" | tr -d "ci" | nl
     1	ssssss
```

```
bchoi@pop-os:~/ex12$ echo "ccccciiiiisssssscis" | tr -d "cs" | nl
     1	iiiii
bchoi@pop-os:~/ex12$ echo "ccccciiiiisssssscis" | tr -d "is" | nl
     1	cccccc
bchoi@pop-os:~/ex12$ echo "ccccciiiiisssssscis112233" | tr -d "cis" | nl
     1	112233
bchoi@pop-os:~/ex12$ echo "ccccciiiiisssssscis112233" | tr -d "cis12" | nl
     1	33
```

Ex12-2 Use regular expression methods to remove only alphabets or only digits.

```
bchoi@pop-os:~/ex12$ echo "ccccciiiiisssssscis112233" | tr -d "a-z" | nl
     1	112233
bchoi@pop-os:~/ex12$ echo "ccccciiiiisssssscis112233" | tr -d "0-9" | nl
     1	ccccciiiiisssssscis
bchoi@pop-os:~/ex12$ echo "what a wonderful world!" | tr -d "a-l" | nl
     1	wt  wonru wor!
```

Ex12-3

```
bchoi@pop-os:~/ex12$ echo "Well done! R2-D2" | tr -d "a-l" | nl
     1	W on! R2-D2
bchoi@pop-os:~/ex12$ echo "Well done! R2-D2" | tr -d "a-z" | nl
     1	W ! R2-D2
bchoi@pop-os:~/ex12$ echo "Well done! R2-D2" | tr -d "RD2" | nl
     1	Well done! -
bchoi@pop-os:~/ex12$ echo "Well done! R2-D2" | tr -d "RD2-" | nl
     1	Well done!
bchoi@pop-os:~/ex12$ echo "Well done! R2-D2" | tr -d "A-Z" | nl
     1	ell done! 2-2
bchoi@pop-os:~/ex12$ echo "Well done! R2-D2" | tr -d "A-R" | nl
     1	Well done! 2-2
bchoi@pop-os:~/ex12$ echo "Well done! R2-D2" | tr -d "A-Z""a-z" | nl
     1	 ! 2-2
bchoi@pop-os:~/ex12$ echo "Well done! R2-D2" | tr -d "A-Za-z!" | nl
     1	  2-2
bchoi@pop-os:~/ex12$ echo "Well done! R2-D2" | tr -d '[:upper:]' | nl
     1	ell done! 2-2
bchoi@pop-os:~/ex12$ echo "Well done! R2-D2" | tr -d '[:lower:]' | nl
     1	W ! R2-D2
bchoi@pop-os:~/ex12$ echo "Well done! R2-D2" | tr -d '[:digit:]' | nl
     1	Well done! R-D
```

Ex12-4 Trimming a number range

```
bchoi@pop-os:~/ex12$ echo {1..10}
1 2 3 4 5 6 7 8 9 10
bchoi@pop-os:~/ex12$ echo {1..10} | tr -d '1-5'
    6 7 8 9 0
bchoi@pop-os:~/ex12$ echo {1..10} | tr -d '3-8'
1 2      9 10
```

Ex12-5 Use tr to replace a character in a string.

```
bchoi@pop-os:~/ex12$ echo "Chloe"
```

```
Chloe
bchoi@pop-os:~/ex12$ echo "Chloe" | tr "C" "K"
Khloe
bchoi@pop-os:~/ex12$ echo "mime"
mime
bchoi@pop-os:~/ex12$ echo "mime" | tr "m" "t"
time
bchoi@pop-os:~/ex12$ echo "mime" | tr "m" "$"
$i$e
bchoi@pop-os:~/ex12$ echo "Ashley" | tr "y" "e"
Ashlee
bchoi@pop-os:~/ex12$ echo "Ashley" | tr "A" "a"
ashley
bchoi@pop-os:~/ex12$ echo "abc" | tr "abc" "ABC"
ABC
bchoi@pop-os:~/ex12$ echo "abc" | tr "abc" "123"
123
bchoi@pop-os:~/ex12$ echo "abcd" | tr "abc" "123"
123d
```

Ex12-6 Practicing what we have learned. Remove parts of a string and mask with "*".

```
bchoi@pop-os:~/ex12$ echo "Brendan 0498765432" | tr "[:digit:]" "*"
Brendan **********
bchoi@pop-os:~/ex12$ echo "Brendan 0498765432" | tr "A-Za-z" "*"
******* 0498765432
bchoi@pop-os:~/ex12$ echo "Brendan 0498765432" | tr "[:lower:][:upper:]" "*"
******* 0498765432
bchoi@pop-os:~/ex12$ echo "Brendan 0498765432" | tr "[:lower:]" "*"
B****** 0498765432
```

Ex12-7 Change lower case to upper case. And upper case to lower case.

```
bchoi@pop-os:~/ex12$ echo "koala"
koala
bchoi@pop-os:~/ex12$ echo "koala" | tr "koala" "KOALA" | nl
     1	KOALA
bchoi@pop-os:~/ex12$ echo "koala" | tr "a-z" "A-Z" | nl
     1	KOALA
bchoi@pop-os:~/ex12$ echo "koala" | tr "[:lower:]" "[:upper:]" | nl
     1	KOALA
bchoi@pop-os:~/ex12$ echo "EMU"
EMU
bchoi@pop-os:~/ex12$ echo "EMU" | tr "EMU" "emu"
emu
bchoi@pop-os:~/ex12$ echo "EMU" | tr "A-Z" "a-z"
emu
bchoi@pop-os:~/ex12$ echo "EMU" | tr "[:upper:]" "[:lower:]"
emu
```

Ex12-8 Use "-s" option, s for squeez

```
bchoi@pop-os:~$ echo "aaaaabbbbbccc" | tr -s a
abbbbbccc
```

```
bchoi@pop-os:~$ echo "aaaaabbbbbccc" | tr -s abc
abc
```

Ex12-9 Change the space with other characters.

```
bchoi@pop-os:~$ echo "London Paris Tokyo Sydney"
London Paris Tokyo Sydney
bchoi@pop-os:~$ echo "London Paris Tokyo Sydney" | tr " " "\n"
London
Paris
Tokyo
Sydney
bchoi@pop-os:~$ echo "London Paris Tokyo Sydney" | tr " " "\t"London  Paris      Tokyo    Sydney
bchoi@pop-os:~$ echo "London Paris Tokyo Sydney" | tr " " "\v"      <<< vertical
London
      Paris
           Tokyo
                Sydney
bchoi@pop-os:~$ echo "London Paris Tokyo Sydney" | tr " " "\012"
London
Paris
Tokyo
Sydney

bchoi@pop-os:~$ who
bchoi    :1          2022-02-26 09:03 (:1)
bchoi@pop-os:~$ who | tr -s " "
bchoi :1 2022-02-26 09:03 (:1)         <<< Easier to view
```

Ex12-10 Check disk usage using "-s" option.

```
bchoi@pop-os:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           775M  1.9M  773M   1% /run
/dev/sda1       106G   96G  5.0G  96% /
tmpfs           3.8G   32K  3.8G   1% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
tmpfs           775M  136K  775M   1% /run/user/1000

bchoi@pop-os:~$ df -h | tr -s " "
Filesystem Size Used Avail Use% Mounted on
tmpfs 775M 1.9M 773M 1% /run
/dev/sda1 106G 96G 5.0G 96% /
tmpfs 3.8G 32K 3.8G 1% /dev/shm
tmpfs 5.0M 4.0K 5.0M 1% /run/lock
tmpfs 775M 136K 775M 1% /run/user/1000
```

Ex12-11 Apply '-s' option.

```
bchoi@pop-os:~$ head -10 /etc/passwd > p1
bchoi@pop-os:~$ nl p1
     1	root:x:0:0:root:/root:/bin/bash
     2	daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
     3	bin:x:2:2:bin:/bin:/usr/sbin/nologin
     4	sys:x:3:3:sys:/dev:/usr/sbin/nologin
     5	sync:x:4:65534:sync:/bin:/bin/sync
     6	games:x:5:60:games:/usr/games:/usr/sbin/nologin
     7	man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
     8	lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
     9	mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
    10	news:x:9:9:news:/var/spool/news:/usr/sbin/nologin

bchoi@pop-os:~$ cat p1 | tr ':' '\t'
root    x       0       0       root    /root   /bin/bash
daemon          x       1       1       daemon          /usr/sbin       /usr/sbin/nologin
bin     x       2       2       bin     /bin    /usr/sbin/nologin
sys     x       3       3       sys     /dev    /usr/sbin/nologin
sync    x       4       65534   sync    /bin    /bin/sync
games   x       5       60      games   /usr/games      /usr/sbin/nologin
man     x       6       12      man     /var/cache/man          /usr/sbin/nologin
lp      x       7       7       lp      /var/spool/lpd  /usr/sbin/nologin
mail    x       8       8       mail    /var/mail       /usr/sbin/nologin
news    x       9       9       news    /var/spool/news         /usr/sbin/nologin

bchoi@pop-os:~$ cat p1 | tr ':' ','
root,x,0,0,root,/root,/bin/bash
daemon,x,1,1,daemon,/usr/sbin,/usr/sbin/nologin
bin,x,2,2,bin,/bin,/usr/sbin/nologin
sys,x,3,3,sys,/dev,/usr/sbin/nologin
sync,x,4,65534,sync,/bin,/bin/sync
games,x,5,60,games,/usr/games,/usr/sbin/nologin
man,x,6,12,man,/var/cache/man,/usr/sbin/nologin
lp,x,7,7,lp,/var/spool/lpd,/usr/sbin/nologin
mail,x,8,8,mail,/var/mail,/usr/sbin/nologin
news,x,9,9,news,/var/spool/news,/usr/sbin/nologin

bchoi@pop-os:~$ cat p1 | tr ':' ',' > p2.csv      <<< save the output to a .csv file.
bchoi@pop-os:~$ more p2.csv
root,x,0,0,root,/root,/bin/bash
daemon,x,1,1,daemon,/usr/sbin,/usr/sbin/nologin
bin,x,2,2,bin,/bin,/usr/sbin/nologin
sys,x,3,3,sys,/dev,/usr/sbin/nologin
sync,x,4,65534,sync,/bin,/bin/sync
games,x,5,60,games,/usr/games,/usr/sbin/nologin
man,x,6,12,man,/var/cache/man,/usr/sbin/nologin
lp,x,7,7,lp,/var/spool/lpd,/usr/sbin/nologin
mail,x,8,8,mail,/var/mail,/usr/sbin/nologin
news,x,9,9,news,/var/spool/news,/usr/sbin/nologin
```

==

Ex12 Summary

# Getting more help on the command.

Use man page to go over a quick summary.
===

Exercise 13: touch

Touch change file timestamps

time: 20 minutes
==

```
> output redirection
> append
< input redirection
<< here document
<<< here string (not often used, but worth noting)
```

#13-1 What we have learned so far.

```
[jdoe@f33sn1 ~]$ mkdir d13 && cd d13
[jdoe@f33sn1 d13]$ date
Sat 26 Feb 2022 11:11:49 AEDT
[jdoe@f33sn1 d13]$ cp /etc/services s
[jdoe@f33sn1 d13]$ cp /etc/passwd p
[jdoe@f33sn1 d13]$ ll
total 684
-rw-r--r--. 1 jdoe jdoe   1810 Feb 26 11:12 p
-rw-r--r--. 1 jdoe jdoe 692252 Feb 26 11:11 s
[jdoe@f33sn1 d13]$ cal > c1         <<< I want to save cal output to a file c1.
[jdoe@f33sn1 d13]$ date > d1
[jdoe@f33sn1 d13]$ who > w1
[jdoe@f33sn1 d13]$ # stdout(1), stdin(0), stderr(2)

[jdoe@f33sn1 d13]$ cal > c2
[jdoe@f33sn1 d13]$ cal > c3
[jdoe@f33sn1 d13]$ ll c*
-rw-r--r--. 1 jdoe jdoe 168 Feb 26 11:12 c1
-rw-r--r--. 1 jdoe jdoe 168 Feb 26 11:14 c2
-rw-r--r--. 1 jdoe jdoe 168 Feb 26 11:15 c3
[jdoe@f33sn1 d13]$ cat c1
   February 2022
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28

[jdoe@f33sn1 d13]$ cat c3
   February 2022
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28
```

```
#13-2 Difference between ">" and ">>".
[jdoe@f33sn1 d13]$ date
Sat 26 Feb 2022 11:16:32 AEDT
[jdoe@f33sn1 d13]$ date > c1
[jdoe@f33sn1 d13]$ nl c1
     1  Sat 26 Feb 2022 11:16:37 AEDT
[jdoe@f33sn1 d13]$ who > c1
[jdoe@f33sn1 d13]$ nl c1
     1  jdoe     pts/0        2022-02-26 10:57 (192.168.56.1)
[jdoe@f33sn1 d13]$ date >> c1
[jdoe@f33sn1 d13]$ nl c1
     1  jdoe     pts/0        2022-02-26 10:57 (192.168.56.1)
     2  Sat 26 Feb 2022 11:17:38 AEDT
[jdoe@f33sn1 d13]$ cal >> c1
[jdoe@f33sn1 d13]$ nl c1
     1  jdoe     pts/0        2022-02-26 10:57 (192.168.56.1)
     2  Sat 26 Feb 2022 11:17:38 AEDT
     3     February 2022
     4  Su Mo Tu We Th Fr Sa
     5         1  2  3  4  5
     6   6  7  8  9 10 11 12
     7  13 14 15 16 17 18 19
     8  20 21 22 23 24 25 26
     9  27 28
    10
[jdoe@f33sn1 d13]$ echo "Hello"
Hello
[jdoe@f33sn1 d13]$ echo "Hello" > c1
[jdoe@f33sn1 d13]$ nl c1
     1  Hello
[jdoe@f33sn1 d13]$ echo "G'day" > c1
[jdoe@f33sn1 d13]$ nl c1
     1  G'day
[jdoe@f33sn1 d13]$ echo "G'day" >> c1
[jdoe@f33sn1 d13]$ echo "G'day" >> c1
[jdoe@f33sn1 d13]$ echo "G'day" >> c1
[jdoe@f33sn1 d13]$ echo "G'day" >> c1
[jdoe@f33sn1 d13]$ nl c1
     1  G'day
     2  G'day
     3  G'day
     4  G'day
     5  G'day
[jdoe@f33sn1 d13]$ echo "..." >> c1
[jdoe@f33sn1 d13]$ nl c1
     1  G'day
     2  G'day
     3  G'day
     4  G'day
     5  G'day
    6  ...

-----------------------------------------------------------------
FILE DESCRIPTOR
# file descriptor, similar to file pointer concept in C language.
stdout (file descriptor=0), standard input (Keyboard)
stdin (file descriptor=1), standard output (Monitor)
stderr (file descriptor=2), standard error (Monitor)

file descriptor:
fopen - opens file
fclose - closes file

^D (Ctrl+D) - end of file
-----------------------------------------------------------------

# We know that the file descriptor 0 is omitted when we run the following command.
[jdoe@f33sn1 d13]$ cat < c1
G'day
G'day
G'day
G'day
G'day
...
[jdoe@f33sn1 d13]$ cat 0< c1
G'day
G'day
G'day
G'day
G'day
...
[jdoe@f33sn1 d13]$ cat < c1
G'day
G'day
G'day
G'day
G'day
...
[jdoe@f33sn1 d13]$ cat 0< c1
G'day
G'day
G'day
G'day
G'day
...
[jdoe@f33sn1 d13]$ who 1> w1
[jdoe@f33sn1 d13]$ who > w1
[jdoe@f33sn1 d13]$ cat 0< c1
G'day
G'day
G'day
```

```
G'day
G'day
...

# When we run a command to see the output, we are sending it to file descriptor 1 by default.
[jdoe@f33sn1 d13]$ date > d2
[jdoe@f33sn1 d13]$ nl d2
     1  Sat 26 Feb 2022 11:26:09 AEDT
[jdoe@f33sn1 d13]$ date 1> d2
[jdoe@f33sn1 d13]$ nl d2
     1  Sat 26 Feb 2022 11:26:19 AEDT
[jdoe@f33sn1 d13]$ myerror              <<< No such command, but I still want to capture the error.
-bash: myerror: command not found
[jdoe@f33sn1 d13]$ myerror > e1
-bash: myerror: command not found
[jdoe@f33sn1 d13]$ nl e1
[jdoe@f33sn1 d13]$ myerror 1> e1         <<< Sending to monitor as standard output, default behaviour.
-bash: myerror: command not found
# Use file descriptor 2 to caputre the error message to a file.
[jdoe@f33sn1 d13]$ myerror 2> e1            <<< Sending it to monitor and hence to a file, e1
[jdoe@f33sn1 d13]$ nl e1
     1  -bash: myerror: command not found

#13-3 ^D (Ctrl+D) - end of file
[jdoe@f33sn1 d13]$
[jdoe@f33sn1 d13]$ cat > h1
hello
konichiwa
bonjour
hola
              <<< Press Ctrl+D, tells the computer that this is the end of the file h1.
[jdoe@f33sn1 d13]$

# If you are not comfortable with the vi editor, you can first work on another text editor and paste the information.

HELLO
KONICHIWA
BONJOUR
HOLA

[jdoe@f33sn1 d13]$ cat > h2
HELLO
KONICHIWA
BONJOUR
HOLA
[jdoe@f33sn1 d13]$ ^C
[jdoe@f33sn1 d13]$ cat h2
HELLO
KONICHIWA
```

```
BONJOUR
HOLA
[jdoe@f33sn1 d13]$ cat h1
hello
konichiwa
bonjour
hola


#13-4 Use of "<<" End document.
[jdoe@f33sn1 d13]$ cat << END > h3
> hello
> konichiwa
> hola
> END
[jdoe@f33sn1 d13]$ cat h3
hello
konichiwa
hola

[jdoe@f33sn1 d13]$ cat << Z > h4
> bye
> sayonara
> adios
> Z
[jdoe@f33sn1 d13]$ cat h4
bye
sayonara
adios
[jdoe@f33sn1 d13]$ cat << END > h4
> bye
> sayonara
> adios
> END
[jdoe@f33sn1 d13]$ cat << END      <<< Just prints out the output to the screen.
> bye
> sayonara
> adios
> END
bye
sayonara
adios

#13-5 Use of touch
[jdoe@f33sn1 d13]$ ll h*
-rw-r--r--. 1 jdoe jdoe 29 Feb 26 11:44 h1   <<< Remember the file creation time
-rw-r--r--. 1 jdoe jdoe 29 Feb 26 11:51 h2
-rw-r--r--. 1 jdoe jdoe 29 Feb 26 11:55 h3
-rw-r--r--. 1 jdoe jdoe 29 Feb 26 11:59 h4
[jdoe@f33sn1 d13]$ touch h1
[jdoe@f33sn1 d13]$ ll h*
```

```
-rw-r--r--. 1 jdoe jdoe 29 Feb 26 20:20 h1   <<< 'touch' command changes the time without changing
the content
-rw-r--r--. 1 jdoe jdoe 29 Feb 26 11:51 h2
-rw-r--r--. 1 jdoe jdoe 29 Feb 26 11:55 h3
-rw-r--r--. 1 jdoe jdoe 29 Feb 26 11:59 h4
[jdoe@f33sn1 d13]$ touch h5                 <<< makes an empty file with 0 byte in size
[jdoe@f33sn1 d13]$ ll h*
-rw-r--r--. 1 jdoe jdoe 29 Feb 26 20:20 h1
-rw-r--r--. 1 jdoe jdoe 29 Feb 26 11:51 h2
-rw-r--r--. 1 jdoe jdoe 29 Feb 26 11:55 h3
-rw-r--r--. 1 jdoe jdoe 29 Feb 26 11:59 h4
-rw-r--r--. 1 jdoe jdoe  0 Feb 26 20:21 h5   <<< file h5
[jdoe@f33sn1 d13]$ touch -t 02261200 h5      <<< Use the '-t' touch option to change the time,
hacker can change the file access time like this.
[jdoe@f33sn1 d13]$ ll h*
-rw-r--r--. 1 jdoe jdoe 29 Feb 26 20:20 h1
-rw-r--r--. 1 jdoe jdoe 29 Feb 26 11:51 h2
-rw-r--r--. 1 jdoe jdoe 29 Feb 26 11:55 h3
-rw-r--r--. 1 jdoe jdoe 29 Feb 26 11:59 h4
-rw-r--r--. 1 jdoe jdoe  0 Feb 26 12:00 h5
```

==

Ex13 Summary

* Understanding of file descriptor. Easier to understand if you understand the OS.
* While studying Linux and programming lanaguage in C derived lanaguages, file descriptors will be mentioned repeatdly.

==

Exercise 14. mkdir & rmdir

time: 10 minutes
==

#14-1

```
[jdoe@f33sn1 d13]$ mkdir d14 && cd d14
[jdoe@f33sn1 d14]$ mkdir d1
[jdoe@f33sn1 d14]$ ls
d1
[jdoe@f33sn1 d14]$ ls -al
total 0
drwxr-xr-x. 3 jdoe jdoe  16 Feb 26 20:40 .    <<< Represents self directory
drwxr-xr-x. 3 jdoe jdoe 155 Feb 26 20:39 ..   <<< Represents parent directory
drwxr-xr-x. 2 jdoe jdoe   6 Feb 26 20:40 d1
[jdoe@f33sn1 d14]$ ls -Al                   <<< If you don't want to see . and .. directories, use the '-Al' option.
total 0
drwxr-xr-x. 2 jdoe jdoe 6 Feb 26 20:40 d1
```

#14-2 Creating multiple directories at once. With mouse clicks, you can only create one folder at a time. With commands, you can create multiple directories at once.

```
[jdoe@f33sn1 d14]$ mkdir d2 d3 d4 d5
[jdoe@f33sn1 d14]$ ls
d1  d2  d3  d4  d5

[jdoe@f33sn1 d14]$ mkdir mydir{1..10}
[jdoe@f33sn1 d14]$ ls
d1  d3  d5      mydir10  mydir3  mydir5  mydir7  mydir9
d2  d4  mydir1  mydir2   mydir4  mydir6  mydir8
```

#14-3 Remove all directories, then create 100 new directories. Again remove all 100 directories.

```
[jdoe@f33sn1 d14]$ ls
d1  d3  d5      mydir10  mydir3  mydir5  mydir7  mydir9
d2  d4  mydir1  mydir2   mydir4  mydir6  mydir8
[jdoe@f33sn1 d14]$ ^C
[jdoe@f33sn1 d14]$ ^C
[jdoe@f33sn1 d14]$ rmdir m*
[jdoe@f33sn1 d14]$ ls
d1  d2  d3  d4  d5
[jdoe@f33sn1 d14]$ rmdir d*
[jdoe@f33sn1 d14]$ ls
[jdoe@f33sn1 d14]$ mkdir d{1..100}
[jdoe@f33sn1 d14]$ ls
d1    d14  d2   d25  d30  d36  d41  d47  d52  d58  d63  d69  d74  d8   d85  d90  d96
d10   d15  d20  d26  d31  d37  d42  d48  d53  d59  d64  d7   d75  d80  d86  d91  d97
d100  d16  d21  d27  d32  d38  d43  d49  d54  d6   d65  d70  d76  d81  d87  d92  d98
d11   d17  d22  d28  d33  d39  d44  d5   d55  d60  d66  d71  d77  d82  d88  d93  d99
d12   d18  d23  d29  d34  d4   d45  d50  d56  d61  d67  d72  d78  d83  d89  d94
d13   d19  d24  d3   d35  d40  d46  d51  d57  d62  d68  d73  d79  d84  d9   d95
[jdoe@f33sn1 d14]$
[jdoe@f33sn1 d14]$ rmdir d{1..100}
```

```
[jdoe@f33sn1 d14]$ ls
[jdoe@f33sn1 d14]$
```

#14-4 Creating a nested directory using the '-p' option.

```
[jdoe@f33sn1 d14]$ mkdir d1/d2
mkdir: cannot create directory ‘d1/d2’: No such file or directory
[jdoe@f33sn1 d14]$ mkdir -p d1/d2          <<< '-p' stands for parents.
[jdoe@f33sn1 d14]$ ls
d1
[jdoe@f33sn1 d14]$ ls d1/
d2
[jdoe@f33sn1 d14]$ mkdir -p d1/d2/d3/d4/d5
[jdoe@f33sn1 d14]$ ls -R                   <<< Use ls with the '-R' option to view sub-directories.
.:
d1

./d1:
d2

./d1/d2:
d3

./d1/d2/d3:
d4

./d1/d2/d3/d4:
d5

./d1/d2/d3/d4/d5:

[jdoe@f33sn1 d14]$ tree        <<< If you installed tree on your Linux, you can use the 'tree'
command.
.
└── d1
    └── d2
        └── d3
            └── d4
                └── d5

5 directories, 0 files
```

#14-5 Deleting nested directories.

```
[jdoe@f33sn1 d14]$ rmdir d1
rmdir: failed to remove 'd1': Directory not empty
[jdoe@f33sn1 d14]$ rmdir -p d1
rmdir: failed to remove 'd1': Directory not empty
[jdoe@f33sn1 d14]$ rmdir -p d1/d2/d3/d4/d5
[jdoe@f33sn1 d14]$ ls
```

```
[jdoe@f33sn1 d14]$

#14-6 Removing a directory with an item.
[jdoe@f33sn1 d14]$ mkdir d1
[jdoe@f33sn1 d14]$ cal > d1/c1
[jdoe@f33sn1 d14]$ nl d1/c1
     1      February 2022
     2  Su Mo Tu We Th Fr Sa
     3         1  2  3  4  5
     4   6  7  8  9 10 11 12
     5  13 14 15 16 17 18 19
     6  20 21 22 23 24 25 26
     7  27 28
     8
[jdoe@f33sn1 d14]$ rmdir d1
rmdir: failed to remove 'd1': Directory not empty
[jdoe@f33sn1 d14]$ rmdir -r d1
rmdir: invalid option -- 'r'
Try 'rmdir --help' for more information.
[jdoe@f33sn1 d14]$ rm -rf d1/c1
[jdoe@f33sn1 d14]$ rmdir d1
[jdoe@f33sn1 d14]$ ls

# However, 'rmdir' is less often used.

[jdoe@f33sn1 d14]$ cal > d2/c2
[jdoe@f33sn1 d14]$ ls -R
.:
d2

./d2:
c2
[jdoe@f33sn1 d14]$ rm -rf d2/          << -r = recursive, deletes all sub-directories too, -f = forceful
[jdoe@f33sn1 d14]$ ls
[jdoe@f33sn1 d14]$

#14-7 Review exercies (no summary required)
[jdoe@f33sn1 d14]$ rm d1 d2 d3
rm: cannot remove 'd1': Is a directory
rm: cannot remove 'd2': Is a directory
rm: cannot remove 'd3': Is a directory
[jdoe@f33sn1 d14]$ rmdir d1 d2 d3
[jdoe@f33sn1 d14]$ ls
[jdoe@f33sn1 d14]$
[jdoe@f33sn1 d14]$ mkdir dir{1..30}
[jdoe@f33sn1 d14]$ ls
dir1   dir12  dir15  dir18  dir20  dir23  dir26  dir29  dir4  dir7
dir10  dir13  dir16  dir19  dir21  dir24  dir27  dir3   dir5  dir8
dir11  dir14  dir17  dir2   dir22  dir25  dir28  dir30  dir6  dir9
[jdoe@f33sn1 d14]$ rmdir dir*     <<< rmdir dir{1..30}
[jdoe@f33sn1 d14]$ ls
```

```
[jdoe@f33sn1 d14]$
```

==

Ex14 Summary

N/A

==

Exercise 15: cp & rm

time: 20 minutes

```
===============================================================================

#15-1
[jdoe@f33sn1 ~]$ mkdir d15 && cd d15
[jdoe@f33sn1 d15]$ pwd
/home/jdoe/d15
[jdoe@f33sn1 d15]$ cp /etc/passwd .          <<< Save passwd file here with the same name.
[jdoe@f33sn1 d15]$ ls
passwd
[jdoe@f33sn1 d15]$ cp /etc/passwd p1
[jdoe@f33sn1 d15]$ ls -lh
total 8.0K
-rw-r--r--. 1 jdoe jdoe 1.8K Feb 26 21:16 p1
-rw-r--r--. 1 jdoe jdoe 1.8K Feb 26 21:15 passwd
[jdoe@f33sn1 d15]$ cp /etc/passwd p2
[jdoe@f33sn1 d15]$ ll
total 12
-rw-r--r--. 1 jdoe jdoe 1810 Feb 26 21:16 p1
-rw-r--r--. 1 jdoe jdoe 1810 Feb 26 21:16 p2
-rw-r--r--. 1 jdoe jdoe 1810 Feb 26 21:15 passwd
[jdoe@f33sn1 d15]$ rm p1
[jdoe@f33sn1 d15]$ ls
p2  passwd
[jdoe@f33sn1 d15]$ rm -i p2                <<< Use '-i' option for interactive deletion. Can prevent
accidents.
rm: remove regular file 'p2'? y
[jdoe@f33sn1 d15]$ rm -i passwd
rm: remove regular file 'passwd'? n
[jdoe@f33sn1 d15]$ ls
passwd

#15-2 Using alias to add '-i' option to rm command. Then unaliasing the rm command.
[jdoe@f33sn1 d15]$ alias rm='rm -i'
[jdoe@f33sn1 d15]$ alias rm
alias rm='rm -i'
[jdoe@f33sn1 d15]$ touch passwd
[jdoe@f33sn1 d15]$ cp passwd p3
[jdoe@f33sn1 d15]$ cp passwd p4
[jdoe@f33sn1 d15]$ cp passwd p5
[jdoe@f33sn1 d15]$ ls
p3  p4  p5  passwd
[jdoe@f33sn1 d15]$ rm p3
rm: remove regular empty file 'p3'? y    <<< Prompted to answer y/n.
[jdoe@f33sn1 d15]$ \rm p4            <<< Use '\' to overwrite the alias and use the default behaviour.
[jdoe@f33sn1 d15]$ rm -f p5
[jdoe@f33sn1 d15]$ ls
passwd
[jdoe@f33sn1 d15]$ rm passwd
rm: remove regular empty file 'passwd'? n  <<< Prompted to answer y/n.
```

```
[jdoe@f33sn1 d15]$ rm -f passwd
[jdoe@f33sn1 d15]$ ls
[jdoe@f33sn1 d15]$
[jdoe@f33sn1 d15]$ unalias rm     <<< Remove the alias
[jdoe@f33sn1 d15]$ alias rm
-bash: alias: rm: not found
```

#15-3 Create a new sub-directory and copy some files. Check the difference betwen 'rmdir' and 'rm -r' commands.

```
[jdoe@f33sn1 d15]$ mkdir sd1
[jdoe@f33sn1 d15]$ cp /etc/a* sd1
cp: -r not specified; omitting directory '/etc/abrt'
cp: -r not specified; omitting directory '/etc/alternatives'
cp: -r not specified; omitting directory '/etc/audit'
cp: -r not specified; omitting directory '/etc/authselect'
[jdoe@f33sn1 d15]$ ls sd1
adjtime  aliases  at.deny
[jdoe@f33sn1 d15]$ rmdir sd1                    <<< rmdir cannot delete a directory with items.
rmdir: failed to remove 'sd1': Directory not empty
[jdoe@f33sn1 d15]$ rm -r sd1                    <<< Use rm with '-r' option to delete a directory with items.
[jdoe@f33sn1 d15]$ ls
```

#15-4 Removing a directory using '-r' (recursive) and '-f' (force)

```
[jdoe@f33sn1 d15]$ mkdir sd2
[jdoe@f33sn1 d15]$ cp /etc/b* sd2
cp: -r not specified; omitting directory '/etc/bash_completion.d'
cp: -r not specified; omitting directory '/etc/binfmt.d'
cp: -r not specified; omitting directory '/etc/bluetooth'
[jdoe@f33sn1 d15]$ ls sd2
bashrc  bindresvport.blacklist
[jdoe@f33sn1 d15]$ rm -rf sd2
[jdoe@f33sn1 d15]$ ls
[jdoe@f33sn1 d15]$
```

#15-5 Copy all items under /etc/ to the current working directory.

```
[jdoe@f33sn1 d15]$ cp /etc/* .                  <<< This command will only copy the files, but the directories.
cp: -r not specified; omitting directory '/etc/abrt'
cp: -r not specified; omitting directory '/etc/alternatives'
cp: -r not specified; omitting directory '/etc/audit'
[...omitted for brevity]
cp: -r not specified; omitting directory '/etc/yum.repos.d'
```

#TIP: If you don't want to see the error messages while copying the files. Add '2>/dev/null' to your command.

```
[jdoe@f33sn1 d15]$ cp /etc/* . 2>/dev/null        <<< Send the messages to '2>/dev/null', to a blackhole.
[jdoe@f33sn1 d15]$
```

```
[jdoe@f33sn1 d15]$ ls
adjtime                issue                  passwd
aliases               issue.net               passwd-
[...omitted for brevity]
inittab               opensc-x86_64.conf      wgetrc
inputrc                os-release             xattr.conf

# Remove files starting with a certain alphabet.

[jdoe@f33sn1 d15]$ rm -f a*
[jdoe@f33sn1 d15]$ rm -f D* G*
[jdoe@f33sn1 d15]$ rm -f b* c* d*        <<< You cannot remove files this way using mouse clicks on
Windows OS.
[jdoe@f33sn1 d15]$ rm -f [g-s]*          <<< Use [g-s] range command to be more efficient.
[jdoe@f33sn1 d15]$ ls
trusted-key.key  updatedb.conf  usb_modeswitch.conf  virc  wgetrc  xattr.conf
[jdoe@f33sn1 d15]$ rm -f *
[jdoe@f33sn1 d15]$ ls

#15-6 Repeat and consolidate your learning.
[jdoe@f33sn1 d15]$ cp /etc/* .
[jdoe@f33sn1 d15]$ rm -f *

[jdoe@f33sn1 d15]$ cp /etc/* . 2>/dev/null
[jdoe@f33sn1 d15]$ rm -f [abcdefghijk]*       <<< Remove all files starting with a, b, c, d, ...., k.
[jdoe@f33sn1 d15]$ rm -f [^uvwx]*            <<< Remove all files except files starting with u, v, w, or
x.
[jdoe@f33sn1 d15]$ ls
updatedb.conf  usb_modeswitch.conf  virc  wgetrc  xattr.conf
[jdoe@f33sn1 d15]$ rm -f [!x]*               <<< Remove all files except files starting with x.
[jdoe@f33sn1 d15]$ ls
xattr.conf
[jdoe@f33sn1 d15]$ rm -f x*
[jdoe@f33sn1 d15]$ ls

-----------------------------------------------------
#TIP - Introduce the concept of Regular Expression. Trying to understand regular expression is the
greatness of IT.
Step 1: Go to https://regex101.com/
Step 2: Copy the ls output of the 'ls -1' command, this will list files per line.
Step 3: Test the following expressions on the web site.
^[a-k].* = same as [abcdefghijk]* in Linux
^[^uvwx].* = same as [^uvwx]*  in Linux
^[!x].* = same as [!x]* in Linux
^[A-Z].+ - match file names starting with upper cases
^[a-z].+ - match file names starting with lower cases
^[a-z]{7}$ - match file names with 7 alphabets
-----------------------------------------------------

#15-7 Check the different colors of files, directories, link files, executables, config files, etc.
```

```
[jdoe@f33sn1 d15]$ ls /etc/
[jdoe@f33sn1 d15]$ rm -f *
[jdoe@f33sn1 d15]$ cp -r /etc/b* .        <<< Use the '-r' option to copy the directories as well.
[jdoe@f33sn1 d15]$ ls
bash_completion.d  bashrc  bindresvport.blacklist  binfmt.d  bluetooth
[jdoe@f33sn1 d15]$ ls -lh
total 8.0K
drwxr-xr-x. 2 jdoe jdoe  116 Feb 26 22:44 bash_completion.d
-rw-r--r--. 1 jdoe jdoe 3.0K Feb 26 22:44 bashrc
-rw-r--r--. 1 jdoe jdoe  447 Feb 26 22:44 bindresvport.blacklist
drwxr-xr-x. 2 jdoe jdoe    6 Feb 26 22:44 binfmt.d
drwxr-xr-x. 2 jdoe jdoe   23 Feb 26 22:44 bluetooth
[jdoe@f33sn1 d15]$ rm -rf b*
[jdoe@f33sn1 d15]$ ls

------------------------------------------------------------------
#WARNING!!!
[jdoe@f33sn1 d15]$ rm -rf * <<< Be careful with the rm command with the '-rf' option, this
command will be devastating if you are logged in as the root user. For normal users, the permissions
will prevent you from deleting important files but have to be careful.
------------------------------------------------------------------

#15-8
[jdoe@f33sn1 d15]$ cal > cal
[jdoe@f33sn1 d15]$ date > date
[jdoe@f33sn1 d15]$ cat cal
   February 2022
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28

[jdoe@f33sn1 d15]$ cat date
Sat 26 Feb 2022 22:47:39 AEDT
[jdoe@f33sn1 d15]$ mkdir sd
[jdoe@f33sn1 d15]$ ll
total 8
-rw-r--r--. 1 jdoe jdoe 168 Feb 26 22:51 cal
-rw-r--r--. 1 jdoe jdoe  30 Feb 26 22:51 date
drwxr-xr-x. 2 jdoe jdoe   6 Feb 26 22:51 sd
[jdoe@f33sn1 d15]$ cp /etc/passwd .
[jdoe@f33sn1 d15]$ cp /etc/passwd passwd2
[jdoe@f33sn1 d15]$ ls
cal  date  passwd  passwd2  sd
[jdoe@f33sn1 d15]$ ll
total 16
-rw-r--r--. 1 jdoe jdoe  168 Feb 26 22:51 cal
-rw-r--r--. 1 jdoe jdoe   30 Feb 26 22:51 date
-rw-r--r--. 1 jdoe jdoe 1810 Feb 26 22:52 passwd
```

```
-rw-r--r--. 1 jdoe jdoe 1810 Feb 26 22:52 passwd2
drwxr-xr-x. 2 jdoe jdoe    6 Feb 26 22:51 sd
[jdoe@f33sn1 d15]$ cp /etc/passwd cal              <<< No prompt for overwriting the file.
[jdoe@f33sn1 d15]$ ll
total 16
-rw-r--r--. 1 jdoe jdoe 1810 Feb 26 22:52 cal       <<< Check the file size. You can see that the file has
been overwritten.
-rw-r--r--. 1 jdoe jdoe   30 Feb 26 22:51 date
-rw-r--r--. 1 jdoe jdoe 1810 Feb 26 22:52 passwd
-rw-r--r--. 1 jdoe jdoe 1810 Feb 26 22:52 passwd2
drwxr-xr-x. 2 jdoe jdoe    6 Feb 26 22:51 sd
[jdoe@f33sn1 d15]$ cp -i /etc/passwd date           <<< Use 'i' option to prevent accidents.
cp: overwrite 'date'? no
[jdoe@f33sn1 d15]$ ll
total 16
-rw-r--r--. 1 jdoe jdoe 1810 Feb 26 22:52 cal
-rw-r--r--. 1 jdoe jdoe   30 Feb 26 22:51 date
-rw-r--r--. 1 jdoe jdoe 1810 Feb 26 22:52 passwd
-rw-r--r--. 1 jdoe jdoe 1810 Feb 26 22:52 passwd2
drwxr-xr-x. 2 jdoe jdoe    6 Feb 26 22:51 sd
```

#TIP - For 'cp' and 'rm' commands, it is best practice to use the 'i' option whenever you run these commands. Reduces the chance of overwriting or removing important files.
You can recover some files with debug file system method but, it is not that easy to recover overwritten or removed files.

#15-9 Copying a file to a directory.

```
[jdoe@f33sn1 d15]$ cp /etc/passwd sd
[jdoe@f33sn1 d15]$ ls sd/
passwd
[jdoe@f33sn1 d15]$ ls -R sd          <<< sd is a directory, so the file is saved as passwd under the sd
directory.
sd:
passwd
[jdoe@f33sn1 d15]$ cp -i /etc/passwd sd/passwd2 <<< You can also give a new name to this file.
[jdoe@f33sn1 d15]$ ls -R sd
sd:
passwd  passwd2
```

# Always check that there are no files with the same name as the 'cp' command will overwrite the existing file, which can cause problems.

#TIP - use 'cp --help' to review the usage

```
Usage: cp [OPTION]... [-T] SOURCE DEST
  or:  cp [OPTION]... SOURCE... DIRECTORY
  or:  cp [OPTION]... -t DIRECTORY SOURCE...
Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.

Mandatory arguments for long options are mandatory for short options too.
```

```
  -a, --archive                same as -dR --preserve=all
      --attributes-only        don't copy the file data, just the attributes
      --backup[=CONTROL]       make a backup of each existing destination file
  -b                           like --backup but does not accept an argument
      --copy-contents          copy contents of special files when recursive
  -d                           same as --no-dereference --preserve=links
  -f, --force                  if an existing destination file cannot be
                                 opened, remove it, and try again (this option
                                 is ignored when the -n option is also used)
  -i, --interactive            prompt before overwrite (overrides a previous -n
                                  option)
[...omitted for brevity]
```

Extra

#15-10 Basic concept of link and unlink. When we run the unlink command to a file, it recalls the system call to remove the file. This is the next-level study concept.

```
[jdoe@f33sn1 d15]$ ls
cal  date  passwd  passwd2  sd
[jdoe@f33sn1 d15]$ unlink cal
[jdoe@f33sn1 d15]$ ls
date  passwd  passwd2  sd
[jdoe@f33sn1 d15]$ unlink date
[jdoe@f33sn1 d15]$ ls
passwd  passwd2  sd
[jdoe@f33sn1 d15]$ unlink sd
unlink: cannot unlink 'sd': Is a directory
[jdoe@f33sn1 d15]$ ll
total 8
-rw-r--r--. 1 jdoe jdoe 1810 Feb 26 22:52 passwd
-rw-r--r--. 1 jdoe jdoe 1810 Feb 26 22:52 passwd2
drwxr-xr-x. 2 jdoe jdoe   35 Feb 26 23:00 sd
```

==

Ex15 Summary

N/A

==

Exercise 16: rename, mv

time: 15 minutes

```
=====================================================================================
```

#16-1 Using 'rename' command to change the name of a file and a directory.

```
[jdoe@f33sn1 ~]$ mkdir d16 && cd d16
[jdoe@f33sn1 d16]$ ls
[jdoe@f33sn1 d16]$ mkdir dir1
[jdoe@f33sn1 d16]$ cal > cal1
[jdoe@f33sn1 d16]$ ll
total 4
-rw-r--r--. 1 jdoe jdoe 168 Feb 27 14:02 cal1
drwxr-xr-x. 2 jdoe jdoe   6 Feb 27 14:02 dir1
[jdoe@f33sn1 d16]$ rename cal1 calendar100 cal1      <<< Change cal1 to 'calendar100' of cal1.
[jdoe@f33sn1 d16]$ ll
total 4
-rw-r--r--. 1 jdoe jdoe 168 Feb 27 14:02 calendar100
drwxr-xr-x. 2 jdoe jdoe   6 Feb 27 14:02 dir1
[jdoe@f33sn1 d16]$ rename dir directory dir1         <<< Change the 'dir' to 'directory' in dir1.
[jdoe@f33sn1 d16]$ ll
total 4
-rw-r--r--. 1 jdoe jdoe 168 Feb 27 14:02 calendar100
drwxr-xr-x. 2 jdoe jdoe   6 Feb 27 14:02 directory1
```

#16-2 Renaming some files beginning with the same alphabet.

```
[jdoe@f33sn1 d16]$ cp /etc/a* .
cp: -r not specified; omitting directory '/etc/abrt'
cp: -r not specified; omitting directory '/etc/alternatives'
cp: -r not specified; omitting directory '/etc/audit'
cp: -r not specified; omitting directory '/etc/authselect'
[jdoe@f33sn1 d16]$ ll
total 12
-rw-r--r--. 1 jdoe jdoe   16 Feb 27 14:08 adjtime
-rw-r--r--. 1 jdoe jdoe 1529 Feb 27 14:08 aliases
-rw-r--r--. 1 jdoe jdoe    1 Feb 27 14:08 at.deny
[jdoe@f33sn1 d16]$ rename a AAA_ a*
[jdoe@f33sn1 d16]$ ll
total 12
-rw-r--r--. 1 jdoe jdoe   16 Feb 27 14:08 AAA_djtime
-rw-r--r--. 1 jdoe jdoe 1529 Feb 27 14:08 AAA_liases
-rw-r--r--. 1 jdoe jdoe    1 Feb 27 14:08 AAA_t.deny
[jdoe@f33sn1 d16]$ rm -rf *                  <<< Removes all files in the current working directory.
```

#16-3 Renaming multiple files stating with a.

```
[jdoe@f33sn1 d16]$ touch a{1..100}
[jdoe@f33sn1 d16]$ ls
a1    a15  a21  a28  a34  a40  a47  a53  a6   a66  a72  a79  a85  a91  a98
a10   a16  a22  a29  a35  a41  a48  a54  a60  a67  a73  a8   a86  a92  a99
a100  a17  a23  a3   a36  a42  a49  a55  a61  a68  a74  a80  a87  a93
a11   a18  a24  a30  a37  a43  a5   a56  a62  a69  a75  a81  a88  a94
a12   a19  a25  a31  a38  a44  a50  a57  a63  a7   a76  a82  a89  a95
```

```
a13   a2   a26  a32  a39  a45  a51  a58  a64  a70  a77  a83  a9   a96
a14   a20  a27  a33  a4   a46  a52  a59  a65  a71  a78  a84  a90  a97
[jdoe@f33sn1 d16]$ rename a AA_ a*
[jdoe@f33sn1 d16]$ ls
AA_1    AA_18  AA_27  AA_36  AA_45  AA_54  AA_63  AA_72  AA_81  AA_90
AA_10   AA_19  AA_28  AA_37  AA_46  AA_55  AA_64  AA_73  AA_82  AA_91
AA_100  AA_2   AA_29  AA_38  AA_47  AA_56  AA_65  AA_74  AA_83  AA_92
AA_11   AA_20  AA_3   AA_39  AA_48  AA_57  AA_66  AA_75  AA_84  AA_93
AA_12   AA_21  AA_30  AA_4   AA_49  AA_58  AA_67  AA_76  AA_85  AA_94
AA_13   AA_22  AA_31  AA_40  AA_5   AA_59  AA_68  AA_77  AA_86  AA_95
AA_14   AA_23  AA_32  AA_41  AA_50  AA_6   AA_69  AA_78  AA_87  AA_96
AA_15   AA_24  AA_33  AA_42  AA_51  AA_60  AA_7   AA_79  AA_88  AA_97
AA_16   AA_25  AA_34  AA_43  AA_52  AA_61  AA_70  AA_8   AA_89  AA_98
AA_17   AA_26  AA_35  AA_44  AA_53  AA_62  AA_71  AA_80  AA_9   AA_99
[jdoe@f33sn1 d16]$ rm -rf *
```

#16-4 Change the file extension of multiple files. Change .txt to .backup.

```
[jdoe@f33sn1 d16]$ touch b{1..30}.txt
[jdoe@f33sn1 d16]$ ls
b10.txt  b14.txt  b18.txt  b21.txt  b25.txt  b29.txt  b4.txt  b8.txt
b11.txt  b15.txt  b19.txt  b22.txt  b26.txt  b2.txt   b5.txt  b9.txt
b12.txt  b16.txt  b1.txt   b23.txt  b27.txt  b30.txt  b6.txt
b13.txt  b17.txt  b20.txt  b24.txt  b28.txt  b3.txt   b7.txt
[jdoe@f33sn1 d16]$ rename .txt .backup b*
[jdoe@f33sn1 d16]$ ls
b10.backup  b15.backup  b1.backup   b24.backup  b29.backup  b5.backup
b11.backup  b16.backup  b20.backup  b25.backup  b2.backup   b6.backup
b12.backup  b17.backup  b21.backup  b26.backup  b30.backup  b7.backup
b13.backup  b18.backup  b22.backup  b27.backup  b3.backup   b8.backup
b14.backup  b19.backup  b23.backup  b28.backup  b4.backup   b9.backup

[jdoe@f33sn1 d16]$ rename --help
```

#16-5 Now change the file name and leave the extension as .backup.

```
[jdoe@f33sn1 d16]$ ls
b10.backup  b15.backup  b1.backup   b24.backup  b29.backup  b5.backup
b11.backup  b16.backup  b20.backup  b25.backup  b2.backup   b6.backup
b12.backup  b17.backup  b21.backup  b26.backup  b30.backup  b7.backup
b13.backup  b18.backup  b22.backup  b27.backup  b3.backup   b8.backup
b14.backup  b19.backup  b23.backup  b28.backup  b4.backup   b9.backup
[jdoe@f33sn1 d16]$ rename b b_backup_ b*
[jdoe@f33sn1 d16]$ ls
b_backup_10.backup  b_backup_18.backup  b_backup_25.backup  b_backup_4.backup
b_backup_11.backup  b_backup_19.backup  b_backup_26.backup  b_backup_5.backup
b_backup_12.backup  b_backup_1.backup   b_backup_27.backup  b_backup_6.backup
b_backup_13.backup  b_backup_20.backup  b_backup_28.backup  b_backup_7.backup
b_backup_14.backup  b_backup_21.backup  b_backup_29.backup  b_backup_8.backup
b_backup_15.backup  b_backup_22.backup  b_backup_2.backup   b_backup_9.backup
b_backup_16.backup  b_backup_23.backup  b_backup_30.backup
b_backup_17.backup  b_backup_24.backup  b_backup_3.backup
```

#16-6 Use the '-v' (verbose) option to show the changes. In Linux, '-V' normally means version, but the small '-v' option indicates verbose.

```
[jdoe@f33sn1 d16]$ rename -v b_backup_ B_BKU_ b*
`b_backup_10.backup' -> `B_BKU_10.backup'
`b_backup_11.backup' -> `B_BKU_11.backup'
`b_backup_12.backup' -> `B_BKU_12.backup'
`b_backup_13.backup' -> `B_BKU_13.backup'
`b_backup_14.backup' -> `B_BKU_14.backup'
`b_backup_15.backup' -> `B_BKU_15.backup'
`b_backup_16.backup' -> `B_BKU_16.backup'
`b_backup_17.backup' -> `B_BKU_17.backup'
`b_backup_18.backup' -> `B_BKU_18.backup'
`b_backup_19.backup' -> `B_BKU_19.backup'
`b_backup_1.backup' -> `B_BKU_1.backup'
`b_backup_20.backup' -> `B_BKU_20.backup'
`b_backup_21.backup' -> `B_BKU_21.backup'
`b_backup_22.backup' -> `B_BKU_22.backup'
`b_backup_23.backup' -> `B_BKU_23.backup'
`b_backup_24.backup' -> `B_BKU_24.backup'
`b_backup_25.backup' -> `B_BKU_25.backup'
`b_backup_26.backup' -> `B_BKU_26.backup'
`b_backup_27.backup' -> `B_BKU_27.backup'
`b_backup_28.backup' -> `B_BKU_28.backup'
`b_backup_29.backup' -> `B_BKU_29.backup'
`b_backup_2.backup' -> `B_BKU_2.backup'
`b_backup_30.backup' -> `B_BKU_30.backup'
`b_backup_3.backup' -> `B_BKU_3.backup'
`b_backup_4.backup' -> `B_BKU_4.backup'
`b_backup_5.backup' -> `B_BKU_5.backup'
`b_backup_6.backup' -> `B_BKU_6.backup'
`b_backup_7.backup' -> `B_BKU_7.backup'
`b_backup_8.backup' -> `B_BKU_8.backup'
`b_backup_9.backup' -> `B_BKU_9.backup'
[jdoe@f33sn1 d16]$ ls
B_BKU_10.backup  B_BKU_18.backup  B_BKU_25.backup  B_BKU_4.backup
B_BKU_11.backup  B_BKU_19.backup  B_BKU_26.backup  B_BKU_5.backup
B_BKU_12.backup  B_BKU_1.backup   B_BKU_27.backup  B_BKU_6.backup
B_BKU_13.backup  B_BKU_20.backup  B_BKU_28.backup  B_BKU_7.backup
B_BKU_14.backup  B_BKU_21.backup  B_BKU_29.backup  B_BKU_8.backup
B_BKU_15.backup  B_BKU_22.backup  B_BKU_2.backup   B_BKU_9.backup
B_BKU_16.backup  B_BKU_23.backup  B_BKU_30.backup
B_BKU_17.backup  B_BKU_24.backup  B_BKU_3.backup
```

#16-7 Use 'mv' to rename a directory.

```
[jdoe@f33sn1 d16]$ ls
B_BKU_10.backup  B_BKU_18.backup  B_BKU_25.backup  B_BKU_4.backup
B_BKU_11.backup  B_BKU_19.backup  B_BKU_26.backup  B_BKU_5.backup
B_BKU_12.backup  B_BKU_1.backup   B_BKU_27.backup  B_BKU_6.backup
B_BKU_13.backup  B_BKU_20.backup  B_BKU_28.backup  B_BKU_7.backup
B_BKU_14.backup  B_BKU_21.backup  B_BKU_29.backup  B_BKU_8.backup
B_BKU_15.backup  B_BKU_22.backup  B_BKU_2.backup   B_BKU_9.backup
```

```
B_BKU_16.backup  B_BKU_23.backup  B_BKU_30.backup
B_BKU_17.backup  B_BKU_24.backup  B_BKU_3.backup
[jdoe@f33sn1 d16]$ mkdir dir16
[jdoe@f33sn1 d16]$ ls
B_BKU_10.backup  B_BKU_18.backup  B_BKU_25.backup  B_BKU_4.backup
B_BKU_11.backup  B_BKU_19.backup  B_BKU_26.backup  B_BKU_5.backup
B_BKU_12.backup  B_BKU_1.backup   B_BKU_27.backup  B_BKU_6.backup
B_BKU_13.backup  B_BKU_20.backup  B_BKU_28.backup  B_BKU_7.backup
B_BKU_14.backup  B_BKU_21.backup  B_BKU_29.backup  B_BKU_8.backup
B_BKU_15.backup  B_BKU_22.backup  B_BKU_2.backup   B_BKU_9.backup
B_BKU_16.backup  B_BKU_23.backup  B_BKU_30.backup  dir16
B_BKU_17.backup  B_BKU_24.backup  B_BKU_3.backup
[jdoe@f33sn1 d16]$ mv dir16 d_16
[jdoe@f33sn1 d16]$ ls
B_BKU_10.backup  B_BKU_18.backup  B_BKU_25.backup  B_BKU_4.backup
B_BKU_11.backup  B_BKU_19.backup  B_BKU_26.backup  B_BKU_5.backup
B_BKU_12.backup  B_BKU_1.backup   B_BKU_27.backup  B_BKU_6.backup
B_BKU_13.backup  B_BKU_20.backup  B_BKU_28.backup  B_BKU_7.backup
B_BKU_14.backup  B_BKU_21.backup  B_BKU_29.backup  B_BKU_8.backup
B_BKU_15.backup  B_BKU_22.backup  B_BKU_2.backup   B_BKU_9.backup
B_BKU_16.backup  B_BKU_23.backup  B_BKU_30.backup  d_16
B_BKU_17.backup  B_BKU_24.backup  B_BKU_3.backup
[jdoe@f33sn1 d16]$ rename d_16 dir_16 d_16
[jdoe@f33sn1 d16]$ ls
B_BKU_10.backup  B_BKU_18.backup  B_BKU_25.backup  B_BKU_4.backup
B_BKU_11.backup  B_BKU_19.backup  B_BKU_26.backup  B_BKU_5.backup
B_BKU_12.backup  B_BKU_1.backup   B_BKU_27.backup  B_BKU_6.backup
B_BKU_13.backup  B_BKU_20.backup  B_BKU_28.backup  B_BKU_7.backup
B_BKU_14.backup  B_BKU_21.backup  B_BKU_29.backup  B_BKU_8.backup
B_BKU_15.backup  B_BKU_22.backup  B_BKU_2.backup   B_BKU_9.backup
B_BKU_16.backup  B_BKU_23.backup  B_BKU_30.backup  dir_16
B_BKU_17.backup  B_BKU_24.backup  B_BKU_3.backup
```

#16-8 Rename the files again and move all files to dir_16.

```
[jdoe@f33sn1 d16]$ rename B_BKU_ bbk_ B*
[jdoe@f33sn1 d16]$ ls
bbk_10.backup  bbk_17.backup  bbk_23.backup  bbk_2.backup   bbk_8.backup
bbk_11.backup  bbk_18.backup  bbk_24.backup  bbk_30.backup  bbk_9.backup
bbk_12.backup  bbk_19.backup  bbk_25.backup  bbk_3.backup   dir_16
bbk_13.backup  bbk_1.backup   bbk_26.backup  bbk_4.backup
bbk_14.backup  bbk_20.backup  bbk_27.backup  bbk_5.backup
bbk_15.backup  bbk_21.backup  bbk_28.backup  bbk_6.backup
bbk_16.backup  bbk_22.backup  bbk_29.backup  bbk_7.backup
[jdoe@f33sn1 d16]$
[jdoe@f33sn1 d16]$ mv bbk* dir_16
[jdoe@f33sn1 d16]$ ls
dir_16
```

#16-9 Move all files in dir_16 to current working directory. Then move a single file to dir_16 directory.

```
[jdoe@f33sn1 d16]$ mv dir_16/* .
```

```
[jdoe@f33sn1 d16]$ ls
bbk_10.backup  bbk_17.backup  bbk_23.backup  bbk_2.backup   bbk_8.backup
bbk_11.backup  bbk_18.backup  bbk_24.backup  bbk_30.backup  bbk_9.backup
bbk_12.backup  bbk_19.backup  bbk_25.backup  bbk_3.backup   dir_16
bbk_13.backup  bbk_1.backup   bbk_26.backup  bbk_4.backup
bbk_14.backup  bbk_20.backup  bbk_27.backup  bbk_5.backup
bbk_15.backup  bbk_21.backup  bbk_28.backup  bbk_6.backup
bbk_16.backup  bbk_22.backup  bbk_29.backup  bbk_7.backup
[jdoe@f33sn1 d16]$ ls dir_16
[jdoe@f33sn1 d16]$ mv bbk_16.backup dir_16
[jdoe@f33sn1 d16]$ ls dir_16/
bbk_16.backup
```

#16-10 Use mv to move the file and change the name while moving the file.

```
[jdoe@f33sn1 d16]$ cp /etc/passwd .
[jdoe@f33sn1 d16]$ cp /etc/passwd pw
[jdoe@f33sn1 d16]$ ls
bbk_10.backup  bbk_18.backup  bbk_24.backup  bbk_30.backup  bbk_9.backup
bbk_11.backup  bbk_19.backup  bbk_25.backup  bbk_3.backup   dir_16
bbk_12.backup  bbk_1.backup   bbk_26.backup  bbk_4.backup   passwd
bbk_13.backup  bbk_20.backup  bbk_27.backup  bbk_5.backup   pw
bbk_14.backup  bbk_21.backup  bbk_28.backup  bbk_6.backup
bbk_15.backup  bbk_22.backup  bbk_29.backup  bbk_7.backup
bbk_17.backup  bbk_23.backup  bbk_2.backup   bbk_8.backup
[jdoe@f33sn1 d16]$ mv pw dir_16/pw300
[jdoe@f33sn1 d16]$ ls
bbk_10.backup  bbk_18.backup  bbk_24.backup  bbk_30.backup  bbk_9.backup
bbk_11.backup  bbk_19.backup  bbk_25.backup  bbk_3.backup   dir_16
bbk_12.backup  bbk_1.backup   bbk_26.backup  bbk_4.backup   passwd
bbk_13.backup  bbk_20.backup  bbk_27.backup  bbk_5.backup
bbk_14.backup  bbk_21.backup  bbk_28.backup  bbk_6.backup
bbk_15.backup  bbk_22.backup  bbk_29.backup  bbk_7.backup
bbk_17.backup  bbk_23.backup  bbk_2.backup   bbk_8.backup
[jdoe@f33sn1 d16]$ ls dir_16/
bbk_16.backup  pw300
```

#16-11 Trying to create a directory or file in a root or system directory.

```
[jdoe@f33sn1 d16]$ ls /
bin   dev  home  lib64  mnt  proc  run   srv  tmp  var
boot  etc  lib   media  opt  root  sbin  sys  usr
[jdoe@f33sn1 d16]$ mkdir /koala
mkdir: cannot create directory ‘/koala’: Permission denied
[jdoe@f33sn1 d16]$ touch /koala
touch: cannot touch '/koala': Permission denied
[jdoe@f33sn1 d16]$ mv passwd /etc
mv: replace '/etc/passwd', overriding mode 0644 (rw-r--r--)? y
mv: cannot move 'passwd' to '/etc/passwd': Permission denied
[jdoe@f33sn1 d16]$ mv passwd /usr
mv: cannot move 'passwd' to '/usr/passwd': Permission denied

[jdoe@f33sn1 d16]$ ls /tmp                         <<< On Linux, you can copy the files to /tmp.
```

```
systemd-private-12414fb41f8b45fa872425dafa950cbc-chronyd.service-kY4m6e
systemd-private-12414fb41f8b45fa872425dafa950cbc-dbus-broker.service-8XCuif
systemd-private-12414fb41f8b45fa872425dafa950cbc-httpd.service-pO8Dyi
systemd-private-12414fb41f8b45fa872425dafa950cbc-ModemManager.service-hJvsxi
systemd-private-12414fb41f8b45fa872425dafa950cbc-php-fpm.service-mSdLeh
systemd-private-12414fb41f8b45fa872425dafa950cbc-systemd-logind.service-SeCowj
systemd-private-12414fb41f8b45fa872425dafa950cbc-systemd-resolved.service-7DNxnh
[jdoe@f33sn1 d16]$ ls
bbk_10.backup  bbk_18.backup  bbk_24.backup  bbk_30.backup  bbk_9.backup
bbk_11.backup  bbk_19.backup  bbk_25.backup  bbk_3.backup   dir_16
bbk_12.backup  bbk_1.backup   bbk_26.backup  bbk_4.backup   passwd
bbk_13.backup  bbk_20.backup  bbk_27.backup  bbk_5.backup
bbk_14.backup  bbk_21.backup  bbk_28.backup  bbk_6.backup
bbk_15.backup  bbk_22.backup  bbk_29.backup  bbk_7.backup
bbk_17.backup  bbk_23.backup  bbk_2.backup   bbk_8.backup
[jdoe@f33sn1 d16]$ mv passwd /tmp/password16
[jdoe@f33sn1 d16]$ ls /tmp/
total 4
-rw-r--r--. 1 jdoe jdoe 1810 Feb 27 14:29 password16           <<< File has been renamed and moved
here.
drwx------. 3 root root   60 Feb 27 13:58 systemd-private-12414fb41f8b45fa872425dafa950cbc-
chronyd.service-kY4m6e
drwx------. 3 root root   60 Feb 27 13:58 systemd-private-12414fb41f8b45fa872425dafa950cbc-dbus-
broker.service-8XCuif
drwx------. 3 root root   60 Feb 27 13:58 systemd-private-12414fb41f8b45fa872425dafa950cbc-
httpd.service-pO8Dyi
drwx------. 3 root root   60 Feb 27 13:58 systemd-private-12414fb41f8b45fa872425dafa950cbc-
ModemManager.service-hJvsxi
drwx------. 3 root root   60 Feb 27 13:58 systemd-private-12414fb41f8b45fa872425dafa950cbc-php-
fpm.service-mSdLeh
drwx------. 3 root root   60 Feb 27 13:58 systemd-private-12414fb41f8b45fa872425dafa950cbc-
systemd-logind.service-SeCowj
drwx------. 3 root root   60 Feb 27 13:58 systemd-private-12414fb41f8b45fa872425dafa950cbc-
systemd-resolved.service-7DNxnh
```

```
#16-12 Move the directory to another directory.
[jdoe@f33sn1 d16]$ ls
bbk_10.backup  bbk_17.backup  bbk_22.backup  bbk_28.backup  bbk_5.backup
bbk_11.backup  bbk_18.backup  bbk_23.backup  bbk_29.backup  bbk_6.backup
bbk_12.backup  bbk_19.backup  bbk_24.backup  bbk_2.backup   bbk_7.backup
bbk_13.backup  bbk_1.backup   bbk_25.backup  bbk_30.backup  bbk_8.backup
bbk_14.backup  bbk_20.backup  bbk_26.backup  bbk_3.backup   bbk_9.backup
bbk_15.backup  bbk_21.backup  bbk_27.backup  bbk_4.backup   dir_16
[jdoe@f33sn1 d16]$ mv dir_16 /tmp
[jdoe@f33sn1 d16]$ ll /tmp
total 4
drwxr-xr-x. 2 jdoe jdoe   80 Feb 27 14:30 dir_16            <<< Directory has been renamed and moved
here.
-rw-r--r--. 1 jdoe jdoe 1810 Feb 27 14:29 password16
```

```
drwx------. 3 root root   60 Feb 27 13:58 systemd-private-12414fb41f8b45fa872425dafa950cbc-chronyd.service-kY4m6e
drwx------. 3 root root   60 Feb 27 13:58 systemd-private-12414fb41f8b45fa872425dafa950cbc-dbus-broker.service-8XCuif
drwx------. 3 root root   60 Feb 27 13:58 systemd-private-12414fb41f8b45fa872425dafa950cbc-httpd.service-pO8Dyi
drwx------. 3 root root   60 Feb 27 13:58 systemd-private-12414fb41f8b45fa872425dafa950cbc-ModemManager.service-hJvsxi
drwx------. 3 root root   60 Feb 27 13:58 systemd-private-12414fb41f8b45fa872425dafa950cbc-php-fpm.service-mSdLeh
drwx------. 3 root root   60 Feb 27 13:58 systemd-private-12414fb41f8b45fa872425dafa950cbc-systemd-logind.service-SeCowj
drwx------. 3 root root   60 Feb 27 13:58 systemd-private-12414fb41f8b45fa872425dafa950cbc-systemd-resolved.service-7DNxnh
```

#16-13 Review exercise. Copy all the configuration files to the current working directory. Then rename the file extension to '.txt'.

```
[jdoe@f33sn1 d16]$ cp /etc/*.conf .
cp: cannot open '/etc/libaudit.conf' for reading: Permission denied
cp: cannot open '/etc/sudo.conf' for reading: Permission denied
[jdoe@f33sn1 d16]$ ls
chrony.conf   krb5.conf      nfsmount.conf      rsyncd.conf
dnsmasq.conf  ld.so.conf     nsswitch.conf      rsyslog.conf
dracut.conf   libuser.conf   opensc.conf        sestatus.conf
fprintd.conf  locale.conf    opensc-x86_64.conf sysctl.conf
host.conf     logrotate.conf passwdqc.conf      updatedb.conf
idmapd.conf   man_db.conf    php-fpm.conf       usb_modeswitch.conf
jwhois.conf   mke2fs.conf    request-key.conf   xattr.conf
kdump.conf    nfs.conf       resolv.conf
[jdoe@f33sn1 d16]$
[jdoe@f33sn1 d16]$ rename .conf .txt *.conf
[jdoe@f33sn1 d16]$ ls
chrony.txt   krb5.txt      nfs.txt           rsyncd.txt
dnsmasq.txt  ld.so.txt     nsswitch.txt      rsyslog.txt
dracut.txt   libuser.txt   opensc.txt        sestatus.txt
fprintd.txt  locale.txt    opensc-x86_64.txt sysctl.txt
host.txt     logrotate.txt passwdqc.txt      updatedb.txt
idmapd.txt   man_db.txt    php-fpm.txt       usb_modeswitch.txt
jwhois.txt   mke2fs.txt    request-key.txt   xattr.txt
kdump.txt    nfsmount.txt  resolv.txt
```

#16-14 Review exercise. Change all .txt files to .py (Python) files.

```
[jdoe@f33sn1 d16]$ rename .txt .py *.txt                <<< Rename multiple files at once.
[jdoe@f33sn1 d16]$ ls
chrony.py   kdump.py     mke2fs.py        php-fpm.py     updatedb.py
dnsmasq.py  krb5.py      nfsmount.py      request-key.py usb_modeswitch.py
dracut.py   ld.so.py     nfs.py           resolv.py      xattr.py
fprintd.py  libuser.py   nsswitch.py      rsyncd.py
host.py     locale.py    opensc.py        rsyslog.py
idmapd.py   logrotate.py opensc-x86_64.py sestatus.py
jwhois.py   man_db.py    passwdqc.py      sysctl.py
```

```
[jdoe@f33sn1 d16]$ mv sysctl.py sysctl.conf                    <<< Rename a single file only.
[jdoe@f33sn1 d16]$ ls sys*
sysctl.conf
```

TIP - Practicing Linux on the Web
https://repl.it   <<< Launch C language option
https://bellard.org/jslinux/ <<< Choose Arch Linux option
https://copy.sh/   <<< Choose Virtual x86, then Arch Linux

===

Ex16 Summary

N/A

===

Exercise 17: seq, factor, numfmt

time: 15 minutes
==

#17-1 Create and move to the correct working directory.

```
[jdoe@f33sn1 ~]$ mkdir d17 && cd d17
[jdoe@f33sn1 d17]$ pwd
/home/jdoe/d17
```

#17-2 Run simple 'seq' commands.

```
[jdoe@f33sn1 d17]$ seq 5
1
2
3
4
5
[jdoe@f33sn1 d17]$ seq 10
1
2
[...omitted for brevity]
9
10
[jdoe@f33sn1 d17]$ seq 1 30
1
2
3
[...omitted for brevity]
28
29
30
[jdoe@f33sn1 d17]$ seq -s ' ' 1 30       <<< Default increment is 1, not shown in this example.
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
[jdoe@f33sn1 d17]$ seq -s ' ' 1 2 30
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
[jdoe@f33sn1 d17]$ seq -s ' ' 1 3 30
1 4 7 10 13 16 19 22 25 28
[jdoe@f33sn1 d17]$ seq -s ' ' 1 4 30
1 5 9 13 17 21 25 29
[jdoe@f33sn1 d17]$ seq -s ' ' 2 2 30
2 4 6 8 10 12 14 16 18 20 22 24 26 28 30
[jdoe@f33sn1 d17]$ seq -s ' ' 3 3 30
3 6 9 12 15 18 21 24 27 30
[jdoe@f33sn1 d17]$ seq -s ' ' 50 -5 1
50 45 40 35 30 25 20 15 10 5
```

#17-3 Adding the 0's to your sequence result.

```
[jdoe@f33sn1 d17]$ seq -s ' ' 100
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36
37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100
[jdoe@f33sn1 d17]$ seq -w -s ' ' 100
```

```
001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023
024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046
047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064 065 066 067 068 069
070 071 072 073 074 075 076 077 078 079 080 081 082 083 084 085 086 087 088 089 090 091 092
093 094 095 096 097 098 099 100

#17-4 Incrementing with decimal interval.
[jdoe@f33sn1 d17]$ seq -s ', ' 1 0.3 10
1.0, 1.3, 1.6, 1.9, 2.2, 2.5, 2.8, 3.1, 3.4, 3.7, 4.0, 4.3, 4.6, 4.9, 5.2, 5.5, 5.8, 6.1, 6.4, 6.7, 7.0, 7.3, 7.6,
7.9, 8.2, 8.5, 8.8, 9.1, 9.4, 9.7, 10.0
[jdoe@f33sn1 d17]$ seq -s ' ' 1 0.5 10
1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5 7.0 7.5 8.0 8.5 9.0 9.5 10.0

#17-5 Use the '-f' option to define trailing digit numbers.
[jdoe@f33sn1 d17]$ seq -f "%.5f" -s ' ' 1 0.4 10
1.00000 1.40000 1.80000 2.20000 2.60000 3.00000 3.40000 3.80000 4.20000 4.60000 5.00000
5.40000 5.80000 6.20000 6.60000 7.00000 7.40000 7.80000 8.20000 8.60000 9.00000 9.40000
9.80000
[jdoe@f33sn1 d17]$ seq -f "%.3f" -s ' ' 1 0.4 10
1.000 1.400 1.800 2.200 2.600 3.000 3.400 3.800 4.200 4.600 5.000 5.400 5.800 6.200 6.600 7.000
7.400 7.800 8.200 8.600 9.000 9.400 9.800
[jdoe@f33sn1 d17]$ seq -f "%.g" -s ' ' 1 0.5 10
1 2 2 2 3 4 4 4 5 6 6 6 7 8 8 8 9 1e+01 1e+01
[jdoe@f33sn1 d17]$ seq -f "%.3e" -s ' ' 1 0.4 10
1.000e+00 1.400e+00 1.800e+00 2.200e+00 2.600e+00 3.000e+00 3.400e+00 3.800e+00 4.200e+00
4.600e+00 5.000e+00 5.400e+00 5.800e+00 6.200e+00 6.600e+00 7.000e+00 7.400e+00 7.800e+00
8.200e+00 8.600e+00 9.000e+00 9.400e+00 9.800e+00

#17-6 Create a shell script and run the sum. The "`" is the key with "~", this is not a single quote.
[jdoe@f33sn1 d17]$ cat > seq1.sh
sum=0
for i in `seq 10`
do
        sum=`expr $sum + $i`
done

echo " \$sum ==> $sum "

[jdoe@f33sn1 d17]$ bash seq2.sh
 $sum ==> 55
```

Let's write this in python.

```
[jdoe@f33sn1 d17]$ cat > seq1.py
sum = 0
for x in range(11):            <<< Make sure you enter 11, Python only calls up to n-1.
   sum = sum + x
print(sum)

[jdoe@f33sn1 d17]$ python3 seq1.py
```

```
55
```

#17-7 What does factor do? It finds you the factors for a given number.

```
[jdoe@f33sn1 d17]$ factor 3
3: 3
[jdoe@f33sn1 d17]$ factor 5
5: 5
[jdoe@f33sn1 d17]$ factor 7
7: 7
[jdoe@f33sn1 d17]$ factor 13
13: 13
[jdoe@f33sn1 d17]$ factor 4
4: 2 2
[jdoe@f33sn1 d17]$ factor 9
9: 3 3
[jdoe@f33sn1 d17]$ factor 10
10: 2 5
[jdoe@f33sn1 d17]$ factor 30
30: 2 3 5
[jdoe@f33sn1 d17]$ factor 50
50: 2 5 5
[jdoe@f33sn1 d17]$ factor 100
100: 2 2 5 5
```

#17-8 Find factors for multiple numbers at once.

```
[jdoe@f33sn1 d17]$ factor 100 1000 10000
100: 2 2 5 5
1000: 2 2 2 5 5 5
10000: 2 2 2 2 5 5 5 5
```

#17-9 What is you do not enter the value after the 'factor' command?

```
[jdoe@f33sn1 d17]$ factor
20
20: 2 2 5
40
40: 2 2 2 5
60
60: 2 2 3 5
[jdoe@f33sn1 d17]$    [press Ctrl+D to exit]
```

#17-10 Number formatting with '--grouping option. This is a fairly new Linux command.

```
[jdoe@f33sn1 d17]$ numfmt 123456789 --grouping
123,456,789
[jdoe@f33sn1 d17]$ numfmt 1234567890123456789 --grouping
1,234,567,890,123,456,789
```

#17-11 Number formatting to express in bytes as in storage.

```
[jdoe@f33sn1 d17]$ numfmt 1 --to=iec
1
```

```
[jdoe@f33sn1 d17]$ numfmt 1024 --to=iec
1.0K
[jdoe@f33sn1 d17]$ numfmt 1234 --to=iec
1.3K
[jdoe@f33sn1 d17]$ numfmt 1234567 --to=iec
1.2M
[jdoe@f33sn1 d17]$ numfmt 1234567890 --to=iec
1.2G
[jdoe@f33sn1 d17]$ numfmt 1234567890123 --to=iec
1.2T
[jdoe@f33sn1 d17]$ numfmt 1234567890123456 --to=iec
1.1P
[jdoe@f33sn1 d17]$ numfmt 1234567890123456789 --to=iec
1.1E
[jdoe@f33sn1 d17]$ numfmt 1234567890123456789012 --to=iec
1.1Z
[jdoe@f33sn1 d17]$ numfmt 1234567890123456789012345 --to=iec
1.1Y
```

URL: https://italchemy.wordpress.com/2021/04/30/storage-units-refresher-for-data-analytics/amp/

```
[jdoe@f33sn1 d17]$ numfmt 1000 --to=iec
1000
[jdoe@f33sn1 d17]$ numfmt 1000 --to=si
1.0K
```

For more information, use

```
[jdoe@f33sn1 d17]$ man numfmt
```

#17-12 Applyint numfmt. Checking out the disks used for each user for notification.

```
[jdoe@f33sn1 d17]$ du /etc
[jdoe@f33sn1 d17]$ du /etc 2>/dev/null
[jdoe@f33sn1 d17]$ du /etc/* 2>/dev/null
[jdoe@f33sn1 d17]$ du /etc/* 2>/dev/null | sort -n
[jdoe@f33sn1 d17]$ du -sh /etc/* 2>/dev/null | sort -n
[jdoe@f33sn1 d17]$ du -sh /etc/* 2>/dev/null | sort -n | numfmt --to=si

[jdoe@f33sn1 d17]$ du -sh /etc/* 2>/dev/null | sort -n | numfmt --to=si | tail -5
numfmt: rejecting suffix in input: ‘1.3M’ (consider using --from)
0 /etc/system-release
0 /etc/system-release-cpe
0 /etc/terminfo
0 /etc/tmpfiles.d
0 /etc/xdg
```

```
------------------------------------------------------------------
[jdoe@f33sn1 ~]$ seq --help
Usage: seq [OPTION]... LAST
  or:  seq [OPTION]... FIRST LAST
  or:  seq [OPTION]... FIRST INCREMENT LAST
Print numbers from FIRST to LAST, in steps of INCREMENT.
```

==

Ex17 Summary

N/A

==

Exercise 18: expr

time: 13 minutes
==

# Used to test expressions against the strings.

#18-1 Perform simple arithmetic using the 'expr' command

```
[jdoe@f33sn1 ~]$ expr 1 + 2
3
[jdoe@f33sn1 ~]$ expr 3 - 2
1
[jdoe@f33sn1 ~]$ expr 4 \* 5      <<< "\" is called escape metacharacter, also called "escaping"
20
[jdoe@f33sn1 ~]$ expr 4 '*' 5
20
[jdoe@f33sn1 ~]$ expr 4 "*" 5
20
[jdoe@f33sn1 ~]$ expr 20 /  5
4
[jdoe@f33sn1 ~]$ expr 10 / 3   <<< Does not print the decimals
3
[jdoe@f33sn1 ~]$ expr 10 % 3   <<< Shows remainder only
1
```

TIP: Metacharacters in Linux and programming
In computing and systems, special characters (metacharacters) are used to help the programming.

* & | > < ( ) ...   <<< Metacharacters with special meanings

* ALL
& background in Linux, in programming languages, means AND
| OR
> Greater
< Less
(Open bracket
) Close bracket

Use 'expr --help' to read more about the expressions.

```
[jdoe@f33sn1 ~]$ expr --help
```

#18-2 Try the comparison expressions.

```
[jdoe@f33sn1 ~]$ expr 3 > 2
[jdoe@f33sn1 ~]$ expr 3 \> 2             <<< Use the escape backslash '\' for greater or less signs
1                                         <<< True is 1
[jdoe@f33sn1 ~]$ expr 3 \> 20
0                                         <<< False is 0
[jdoe@f33sn1 ~]$ expr 3 \< 20
1
[jdoe@f33sn1 ~]$ expr 3 = 2
```

```
0
[jdoe@f33sn1 ~]$ expr 3 = 3
1
[jdoe@f33sn1 ~]$ expr 3 != 2
1
[jdoe@f33sn1 ~]$ expr 3 == 2              <<< '==' can be used as well.
0
[jdoe@f33sn1 ~]$ expr 3 \<= 2
0
[jdoe@f33sn1 ~]$ expr 3 \>= 2
1
[jdoe@f33sn1 ~]$ expr 3 \>= 3
1
```

TIP: How to check if your last command was valid?
You can check if your last command was correct or incorrect by using 'echo $?'

```
[jdoe@f33sn1 ~]$ hostname
f33sn1
[jdoe@f33sn1 ~]$ echo $?                  <<< Validates the last command, 0 means valid, other
number means invalid
0
[jdoe@f33sn1 ~]$ namehost
-bash: namehost: command not found
[jdoe@f33sn1 ~]$ echo $?
127
```

#18-3 Practice the '|' (OR) operator with different combinations.

```
[jdoe@f33sn1 ~]$ expr 3 \| 2
3
[jdoe@f33sn1 ~]$ expr 3 \| 5
3
[jdoe@f33sn1 ~]$ expr 3 \| 50
3
[jdoe@f33sn1 ~]$ expr 3 \| 0              <<< Enter 0 as the second integer and it works fine.
3
[jdoe@f33sn1 ~]$ expr 3 \| ''             <<< Enter '' (null) and it still works fine.
3
```

# If the first value is 0 or '' (null), the second number will be printed in the OR computation.

```
[jdoe@f33sn1 ~]$ expr 0 \| 3              <<< Change the expression and see what happens
3
[jdoe@f33sn1 ~]$ expr 0 \| 30
30
[jdoe@f33sn1 ~]$ expr '' \| 3
3
[jdoe@f33sn1 ~]$ expr '' \| 30
30
```

#18-4 Practice the '&' (AND) operator. When one expression is 0 or '' (null), it always returns 0.

```
[jdoe@f33sn1 ~]$ expr  3 \& 5
3
[jdoe@f33sn1 ~]$ expr  5 \& 3
5
[jdoe@f33sn1 ~]$ expr  3 \& 0
0
[jdoe@f33sn1 ~]$ expr  0 \& 5
0
[jdoe@f33sn1 ~]$ expr  3 \& ''
0
[jdoe@f33sn1 ~]$ expr  '' \& 5
0
```

This is not used a lot, probably the bit calculation operator is used more often in programming. But the concept is the same and it is always good to build the right foundation after understanding the concepts.

#18-5 Using the 'match' command with the expr command.

```
[jdoe@f33sn1 ~]$ expr match pancake pan
3
[jdoe@f33sn1 ~]$ expr match pancake panca
5
[jdoe@f33sn1 ~]$ expr match pancake pancake
7
[jdoe@f33sn1 ~]$ expr match pancake cake
0                                                   <<< Only matches from the beginning of the string, returns 0 for False

# You can use "STRING | REGEXP" to get the same result as 'match'
[jdoe@f33sn1 ~]$ expr pancake : pan
3
[jdoe@f33sn1 ~]$ expr pancake : pancake
7
```

#18-6 Using 'substr' with expr. You can grab the exact range of characters from the string.

```
[jdoe@f33sn1 ~]$ expr substr pancake 1 3
pan
[jdoe@f33sn1 ~]$ expr substr pancake 1 7
pancake
[jdoe@f33sn1 ~]$ expr substr pancake 4 7
cake
```

#18-7 Use 'expr' to locate the index of a character in a string.

```
[jdoe@f33sn1 ~]$ expr index pancake p
1
[jdoe@f33sn1 ~]$ expr index pancake c
```

```
4
[jdoe@f33sn1 ~]$ expr index pancake k
6
[jdoe@f33sn1 ~]$ expr index pancake z
0                                        <<< If the character is not found in the string, returns 0.

# Works the same way on integers.
[jdoe@f33sn1 ~]$ expr index 12345 1
1
[jdoe@f33sn1 ~]$ expr index 12345 3
3
[jdoe@f33sn1 ~]$ expr index 12345 0
0
[jdoe@f33sn1 ~]$ expr index 12345 6
0
[jdoe@f33sn1 ~]$ expr index 012345 0
1
[jdoe@f33sn1 ~]$ expr index 123450 0
6

#18-8 Use 'length' to find the length of a string.
[jdoe@f33sn1 ~]$ expr length pan
3
[jdoe@f33sn1 ~]$ expr length cake
4
[jdoe@f33sn1 ~]$ expr length pancake
7
```

```
-----------------------------------------------------------------
Use 'expr --help' to view and read all the details about the 'expr' command.
-----------------------------------------------------------------
```

==

Ex18 Summary

N/A

==

Exercise 19: let (an internal command)

time: 14 minutes

```
==============================================================================

[jdoe@f33sn1 ~]$ help let                    <<< use 'help' at the beginning for internal command help

a=3
$a++            a = a+1
$a--            a = a-1

$++a
$--a

!               Not
~               bit not (flip 0 to 1, 1 to 0)
2 * 2
2 ** 3          exponent

a = a + 1       a += 1
----------------------------------------------------------------
```

#19-1 Working with variables and printing the calculated variables.

```
[jdoe@f33sn1 ~]$ a=10
[jdoe@f33sn1 ~]$ a = 10
-bash: a: command not found
[jdoe@f33sn1 ~]$ a=10
[jdoe@f33sn1 ~]$ echo $a
10
[jdoe@f33sn1 ~]$ a=10 && echo $a
10
[jdoe@f33sn1 ~]$ let b=$a && echo $b
10
```

#19-2 Arithematics calculation and printing the output using the 'let' command.

```
[jdoe@f33sn1 ~]$ let b=$++a && echo $b
-bash: let: b=$++a: syntax error: operand expected (error token is "$++a")
[jdoe@f33sn1 ~]$ let b=++a && echo $b
11
[jdoe@f33sn1 ~]$ let b=++a && echo $b
12
[jdoe@f33sn1 ~]$ let b=a++ && echo $b
12
[jdoe@f33sn1 ~]$ echo $a
13
[jdoe@f33sn1 ~]$ let b=a++ && echo $b
13
[jdoe@f33sn1 ~]$ echo $a
14
```

#19-3 Using the power '**' with the let command.

```
[jdoe@f33sn1 ~]$ let b=2*3 && echo $b
```

```
6
[jdoe@f33sn1 ~]$ let b=2**3 && echo $b
8
[jdoe@f33sn1 ~]$ let b=2**10 && echo $b
1024
[jdoe@f33sn1 ~]$ let b=2 ** 10 && echo $b
2
[jdoe@f33sn1 ~]$ let b=(2 ** 10) && echo $b
1024

#19-4 Equals, AND and OR operation with 'let' command.
[jdoe@f33sn1 ~]$ let b=(1 == 10) && echo $b     <<< no result is printed if wrong
[jdoe@f33sn1 ~]$ let b=(1 != 10) && echo $b     <<< Prints 1 (True) if correct.
1
[jdoe@f33sn1 ~]$ let b=(1 & 1) && echo $b
-bash: syntax error near unexpected token `&'
[jdoe@f33sn1 ~]$ let b=(1 \& 1) && echo $b
1
[jdoe@f33sn1 ~]$ let b=(1 \& 0) && echo $b
[jdoe@f33sn1 ~]$ let b=(0 \& 0) && echo $b      <<< no result is printed if 0 or null.

[jdoe@f33sn1 ~]$ let b=(1 \| 0) && echo $b
1
[jdoe@f33sn1 ~]$ let b=(0 \| 0) && echo $b
[jdoe@f33sn1 ~]$ let b=(0 \| 1) && echo $b
1

#19-5 Assigning different values to a variable based on a condition.
[jdoe@f33sn1 ~]$ let a=(3\<5)?100:200           <<< Don't forget the '\' escape character.
[jdoe@f33sn1 ~]$ echo $a
100
[jdoe@f33sn1 ~]$ let a=(3\>5)?100:200
[jdoe@f33sn1 ~]$ echo $a
200

#19-6 Simple calculation using variable and round brackets '(( ))'
[jdoe@f33sn1 ~]$ b=$((2 + 4))
[jdoe@f33sn1 ~]$ echo $b
6
[jdoe@f33sn1 ~]$ b=$((2 + 4)) && echo $b        <<< 'let' can be replaced with the use of the double
round brackets '(( ))'
6
[jdoe@f33sn1 ~]$ b=$((2 * 4)) && echo $b
8
[jdoe@f33sn1 ~]$ b=$((20 / 4)) && echo $b
5
[jdoe@f33sn1 ~]$ b=$((2 ** 10)) && echo $b
1024

#19-7 Shift calculation
[jdoe@f33sn1 ~]$ b=$((2 << 1)) && echo $b
```

```
4
[jdoe@f33sn1 ~]$ b=$((2 << 2)) && echo $b
8
[jdoe@f33sn1 ~]$ b=$((2 << 3)) && echo $b
16
[jdoe@f33sn1 ~]$ b=$((2 << 4)) && echo $b
32

[jdoe@f33sn1 ~]$ b=$((2 >> 4)) && echo $b
0
[jdoe@f33sn1 ~]$ b=$((100 >> 4)) && echo $b
6
[jdoe@f33sn1 ~]$ b=$((100 >> 3)) && echo $b
12
[jdoe@f33sn1 ~]$ b=$((100 >> 2)) && echo $b
25
[jdoe@f33sn1 ~]$ b=$((100 >> 1)) && echo $b
50

#19-8 You can replace the double round brackets with single square brackets.
jdoe@f33sn1 ~]$ c=$[10 + 20] && echo $c                    <<< square brackets can replace 'let' or
double round brackets '(( ))'
30
[jdoe@f33sn1 ~]$ c=$[10 - 20] && echo $c
-10
[jdoe@f33sn1 ~]$ c=$[10 * 20] && echo $c
200
[jdoe@f33sn1 ~]$ c=$[10 / 20] && echo $c
0
[jdoe@f33sn1 ~]$ c=$[10 % 20] && echo $c
10
[jdoe@f33sn1 ~]$ c=$[10 ** 20] && echo $c
7766279631452241920
```

TIP: External vs Internal commands, what is the difference?
External commands are add-ons.
Internal commands work out of the box, included as part of the Linux OS.

```
# Use the 'which' command to locate the location of the external Linux command.
[jdoe@f33sn1 ~]$ which expr
/usr/bin/expr
[jdoe@f33sn1 ~]$ which let                <<< 'let' is not found under '/usr/bin/'
/usr/bin/which: no let in
(/home/jdoe/.local/bin:/home/jdoe/bin:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin)

# Check internal Linux command using 'enable'
[jdoe@f33sn1 ~]$ enable | nl
     1  enable .
     2  enable :
     3  enable [
[...omitted for brevity]
   33  enable kill
   34  enable let            <<< Here is 'let' as an internal command
   35  enable local
[...omitted for brevity]
   59  enable unalias
   60  enable unset
   61  enable wait

# Also, you can use the 'compgen' command to view all the internal Linux commands.
[jdoe@f33sn1 ~]$ compgen -b | column
.           compopt     fg          pushd       trap
:           continue    getopts     pwd         true
[           declare     hash        read        type
alias       dirs        help        readarray   typeset
bg          disown      history     readonly    ulimit
bind        echo        jobs        return      umask
break       enable      kill        set         unalias
builtin     eval        let         shift       unset
caller      exec        local       shopt       wait
cd          exit        logout      source
command     export      mapfile     suspend
compgen     false       popd        test
complete    fc          printf      times
-----------------------------------------------------------------
```

# These are all Shell commands, but what really is a shell?
"shell basically is an interpreter for commands"
Shells are installed out of the box and there are many different types of shells used.

sh, ksh, csh, tcsh, bash (Linux default), zsh, dash (Dash shell)

```
# You can view which shell you are using by running the following command.
[jdoe@f33sn1 ~]$ echo $SHELL
/bin/bash
[jdoe@f33sn1 ~]$ choe            <<< Incorrect command, bash shell interpret to return the command
result
-bash: choe: command not found
```

==

Ex19 Summary

N/A

==