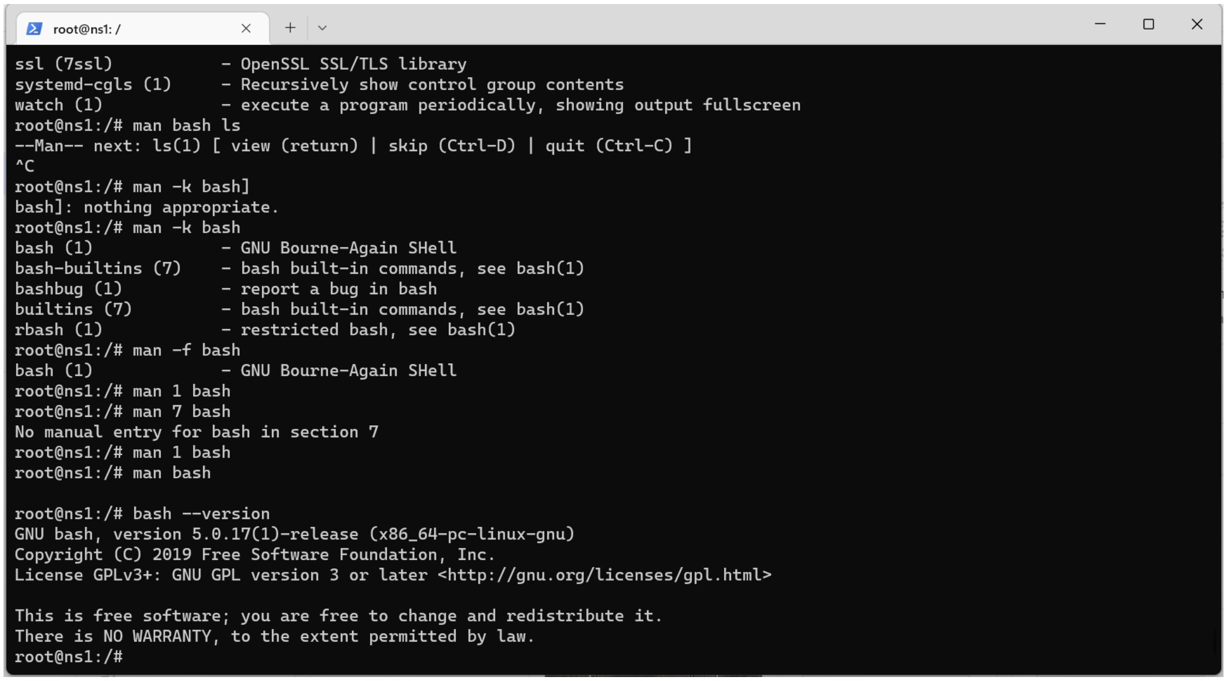
Linux 1

1. Using the `man` command, read about the `-k` and `-f` switches for the `man` command and explain how these might be useful when searching for commands.

The -f command would be useful because it would display a short description thats is similar while the -k command would search the short manual page descriptions for keywords and display any matches.

2. Read the documentation for the `bash` shell using the `man` command and determine how to have `bash` report the version. Show a screenshot of the command used to check the version of `bash` installed on your VM.

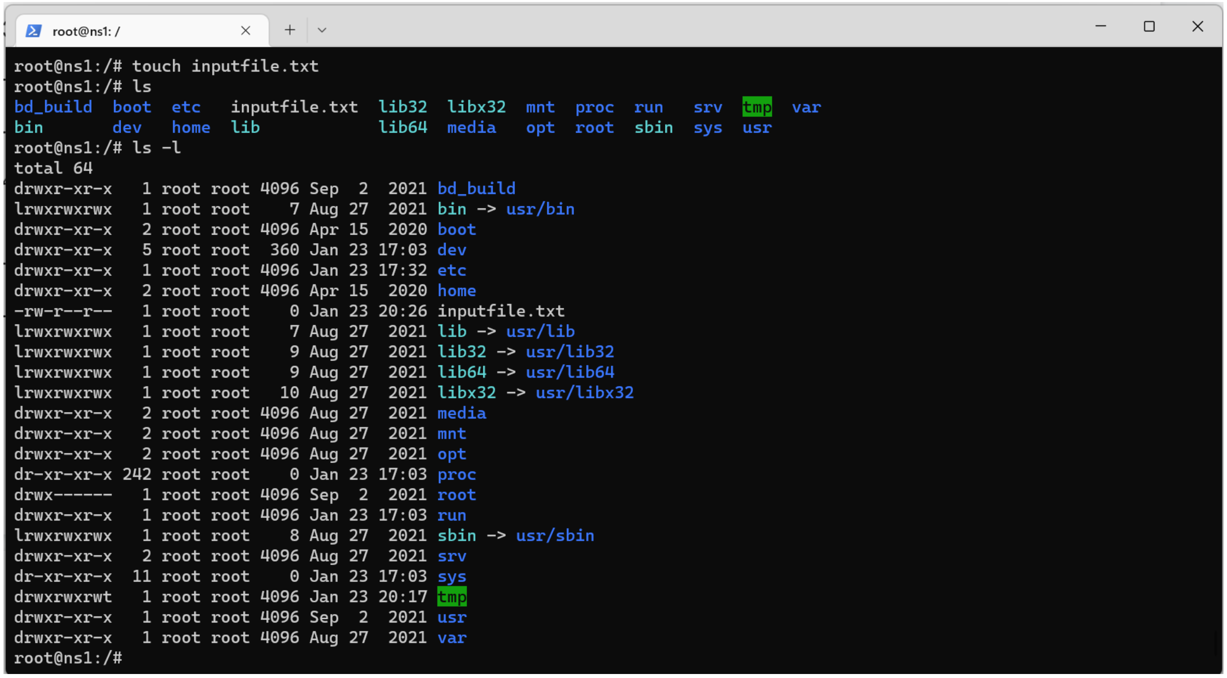


```
ssl (7ssl)           - OpenSSL SSL/TLS library
systemd-cgls (1)     - Recursively show control group contents
watch (1)            - execute a program periodically, showing output fullscreen
root@ns1:/# man bash ls
--Man-- next: ls(1) [ view (return) | skip (Ctrl-D) | quit (Ctrl-C) ]
^C
root@ns1:/# man -k bash]
bash]: nothing appropriate.
root@ns1:/# man -k bash
bash (1)             - GNU Bourne-Again SHell
bash-builtins (7)    - bash built-in commands, see bash(1)
bashbug (1)          - report a bug in bash
builtins (7)         - bash built-in commands, see bash(1)
rbash (1)            - restricted bash, see bash(1)
root@ns1:/# man -f bash
bash (1)             - GNU Bourne-Again SHell
root@ns1:/# man 1 bash
root@ns1:/# man 7 bash
No manual entry for bash in section 7
root@ns1:/# man 1 bash
root@ns1:/# man bash

root@ns1:/# bash --version
GNU bash, version 5.0.17(1)-release (x86_64-pc-linux-gnu)
Copyright (C) 2019 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://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.
root@ns1:/#
```

3. Read the documentation for the `touch` command. Describe what `touch` does.

The touch command allows you to change file timestamps and in some cases create files

4. Use the `touch` command to create a file named `input file.txt`. Include a screenshot of the `touch` command and also the output of `ls -l` showing the newly created file.



```
root@ns1:/# touch inputfile.txt
root@ns1:/# ls
bd_build  boot  etc   inputfile.txt  lib32  libx32  mnt  proc  run   srv  tmp  var
bin       dev   home  lib            lib64  media   opt  root  sbin  sys  usr
root@ns1:/# ls -l
total 64
drwxr-xr-x   1 root root 4096 Sep  2  2021 bd_build
lrwxrwxrwx   1 root root    7 Aug 27  2021 bin -> usr/bin
drwxr-xr-x   2 root root 4096 Apr 15  2020 boot
drwxr-xr-x   5 root root  360 Jan 23 17:03 dev
drwxr-xr-x   1 root root 4096 Jan 23 17:32 etc
drwxr-xr-x   2 root root 4096 Apr 15  2020 home
-rw-r--r--   1 root root    0 Jan 23 20:26 inputfile.txt
lrwxrwxrwx   1 root root    7 Aug 27  2021 lib -> usr/lib
lrwxrwxrwx   1 root root    9 Aug 27  2021 lib32 -> usr/lib32
lrwxrwxrwx   1 root root    9 Aug 27  2021 lib64 -> usr/lib64
lrwxrwxrwx   1 root root   10 Aug 27  2021 libx32 -> usr/libx32
drwxr-xr-x   2 root root 4096 Aug 27  2021 media
drwxr-xr-x   2 root root 4096 Aug 27  2021 mnt
drwxr-xr-x   2 root root 4096 Aug 27  2021 opt
dr-xr-xr-x 242 root root    0 Jan 23 17:03 proc
drwx------   1 root root 4096 Sep  2  2021 root
drwxr-xr-x   1 root root 4096 Jan 23 17:03 run
lrwxrwxrwx   1 root root    8 Aug 27  2021 sbin -> usr/sbin
drwxr-xr-x   2 root root 4096 Aug 27  2021 srv
dr-xr-xr-x  11 root root    0 Jan 23 17:03 sys
drwxrwxrwt   1 root root 4096 Jan 23 20:17 tmp
drwxr-xr-x   1 root root 4096 Sep  2  2021 usr
drwxr-xr-x   1 root root 4096 Aug 27  2021 var
root@ns1:/#
```

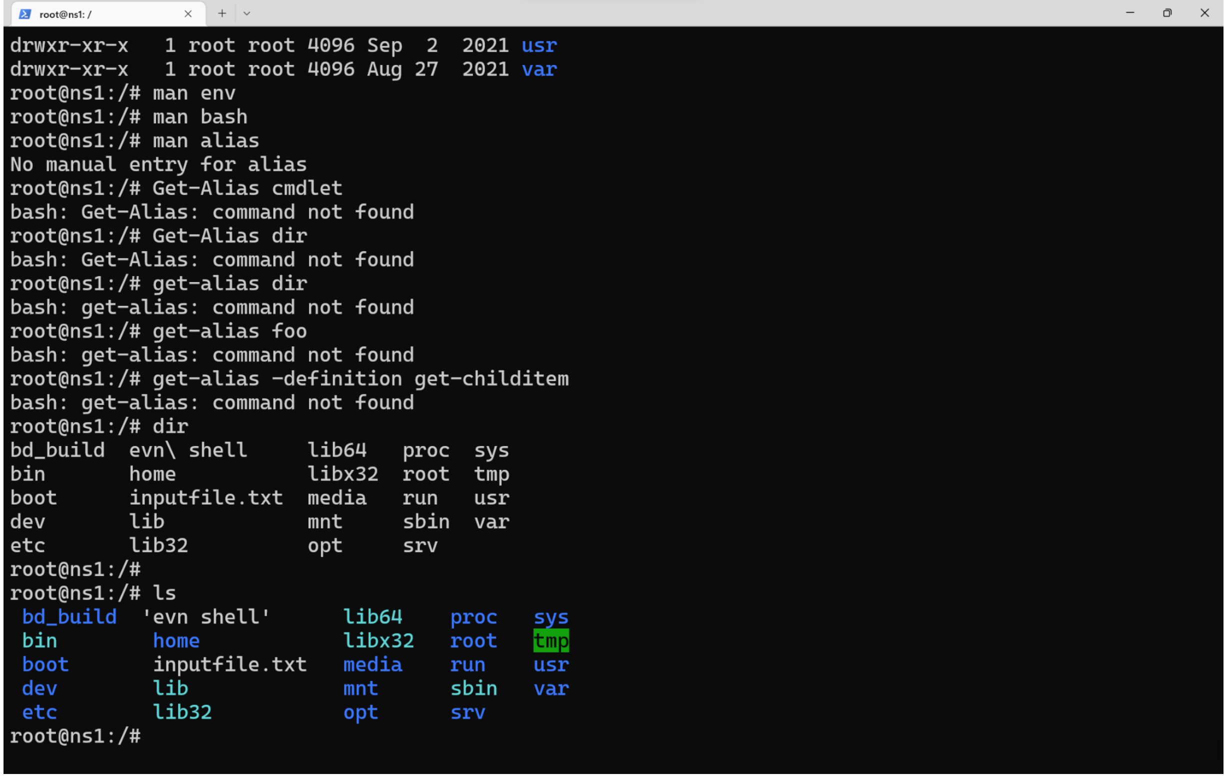
5. Using the `env` command, determine the value of the `SHELL` variable. What does this variable value indicate? Use the `bash` documentation to find this out.

The full pathname to the shell is kept in this environment variable. If it is not set when the shell starts, bash assigns to it the full pathname of the current user's login shell.

6. What does the variable `SHLVL` indicate? Use the `bash` documentation to find this?

The variable SHLVL indicates an increment by one each time an instance of bash is started.

7. Show a screenshot of the `alias` command and describe what aliases use the `ls` command.



```
drwxr-xr-x   1 root root 4096 Sep  2  2021 usr
drwxr-xr-x   1 root root 4096 Aug 27  2021 var
root@ns1:/# man env
root@ns1:/# man bash
root@ns1:/# man alias
No manual entry for alias
root@ns1:/# Get-Alias cmdlet
bash: Get-Alias: command not found
root@ns1:/# Get-Alias dir
bash: Get-Alias: command not found
root@ns1:/# get-alias dir
bash: get-alias: command not found
root@ns1:/# get-alias foo
bash: get-alias: command not found
root@ns1:/# get-alias -definition get-childitem
bash: get-alias: command not found
root@ns1:/# dir
bd_build  evn\ shell     lib64   proc  sys
bin       home           libx32  root  tmp
boot      inputfile.txt  media   run   usr
dev       lib            mnt     sbin  var
etc       lib32          opt     srv
root@ns1:/#
root@ns1:/# ls
 bd_build  'evn shell'       lib64    proc   sys
 bin        home             libx32   root   tmp
 boot       inputfile.txt    media    run    usr
 dev        lib              mnt      sbin   var
 etc        lib32            opt      srv
root@ns1:/#
```

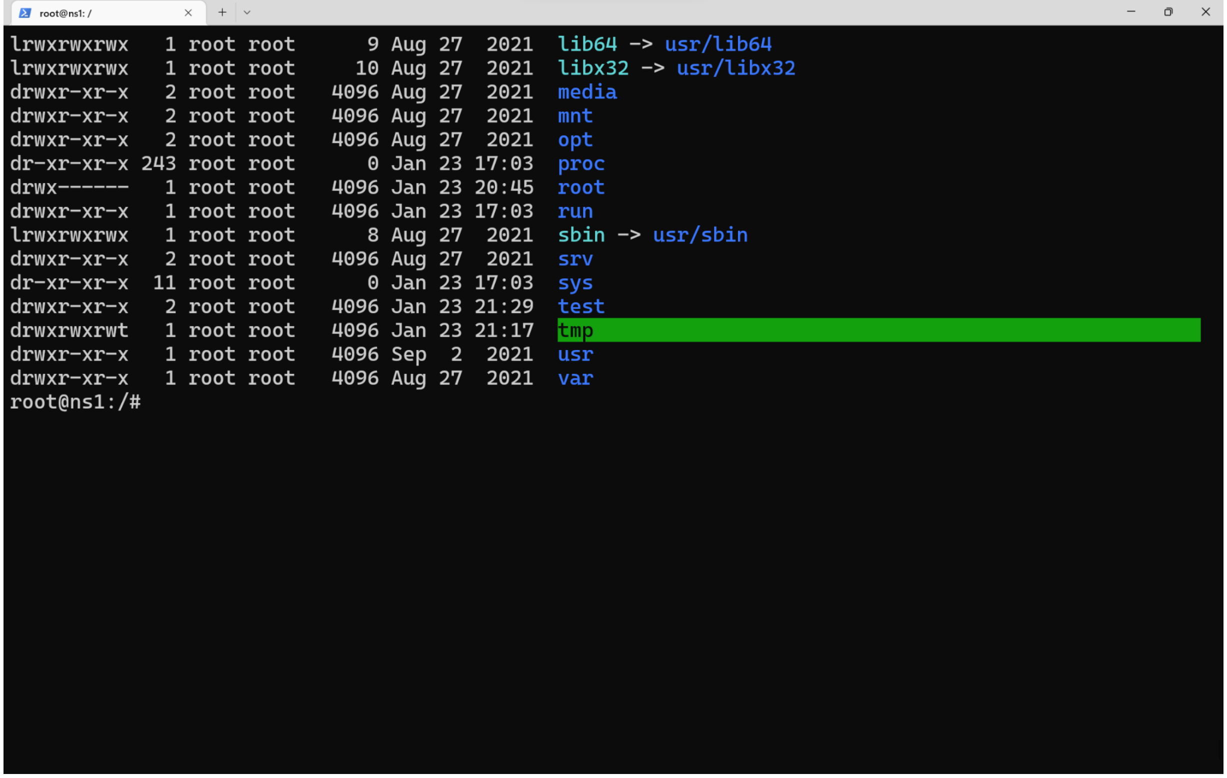
Aliases used by the ls command are gci and dir

8. Pick one alias that uses `ls` and describe the switches used by the alias. You will need to read the documentation for the `ls` command to determine what the switches do.

Switches include —all which do not ignore entries starting with ‘.’, –author while printing the author of each file, -c which lists entries by columns, and -d which lists directories themselves, not their contents.

9. Create a directory named `test` using the `mkdir` command. What are the permissions that the `test` directory has? You can examine permissions using the `ls -l` command. Include a screenshot of the output of the command `ls -l` with your explanation.

Permissions include drwxr-xr-x which means the directory can be changed, navigated, and read by the owner, navigated and read by the group, and navigated by others



```
lrwxrwxrwx   1 root root     9 Aug 27  2021 lib64 -> usr/lib64
lrwxrwxrwx   1 root root    10 Aug 27  2021 libx32 -> usr/libx32
drwxr-xr-x   2 root root  4096 Aug 27  2021 media
drwxr-xr-x   2 root root  4096 Aug 27  2021 mnt
drwxr-xr-x   2 root root  4096 Aug 27  2021 opt
dr-xr-xr-x 243 root root     0 Jan 23 17:03 proc
drwx------   1 root root  4096 Jan 23 20:45 root
drwxr-xr-x   1 root root  4096 Jan 23 17:03 run
lrwxrwxrwx   1 root root     8 Aug 27  2021 sbin -> usr/sbin
drwxr-xr-x   2 root root  4096 Aug 27  2021 srv
dr-xr-xr-x  11 root root     0 Jan 23 17:03 sys
drwxr-xr-x   2 root root  4096 Jan 23 21:29 test
drwxrwxrwt   1 root root  4096 Jan 23 21:17 tmp
drwxr-xr-x   1 root root  4096 Sep  2  2021 usr
drwxr-xr-x   1 root root  4096 Aug 27  2021 var
root@ns1:/#
```

10. What are the permissions of the file `input file.txt` that you created above?

The permissions for the file input file.txt is -rw-r–r– meaning the file can be read and modified by the owner, read by the group, and read by others

11. What files do you think are created from running the following command?

```
./my_fancy_script.sh 1>output.txt 2>error.log
```

A script file named my_fancy_script was created

12. Read the documentation for `cat` and `less`. Is there a difference between the two commands listed below? If so, explain what that difference is. If not, describe how to remove the use of the `cat` command.

```
cat /etc/passwd | less

less /etc/passwd
```

Cat allows you to concatenate files and print on the standard output while less command lets you view the content of the input. The difference between cat and less is that less is for file reading that lets you scroll through it. Cat is string manipulation.

13. After going through the vimtutor tutorial, share your thoughts on the Vim editor

The vim editor seems very quick and efficient that allows you to use it while using the terminal and lets you edit the files while you're using the terminal. Vim also is cross platform. Vim seems super simple and the navigation controls are straight forward

14. After going through the Emacs guided tour, share your thoughts on the Emacs editor

There seems to be some more tedious button controls to navigate through the text editor. There seems to be more of a learning curve with using this text editor because you need to remember a lot of different key combos to perform certain features.

15. After going through the tutorial for nano, share your thoughts on the nano editor

Navigation and editing is super simple and doesn't require you to remember any key combos. Any controls that you need, although limited, are available in plain view for the user.

16. Which of the editors do you think you will use for the semester when editing scripts and configuration files?

I would use nano because of how straightforward and consistent it is.