What Is grep?

Grep is a Linux command-line tool that allows users to search files for a specified textual pattern. When **grep** finds a match, it prints lines containing that pattern to the terminal.

By default, the **grep** command outputs entire lines that contain the match. Users can utilize **grep** options to include additional context around the match or show only matching parts of the line.

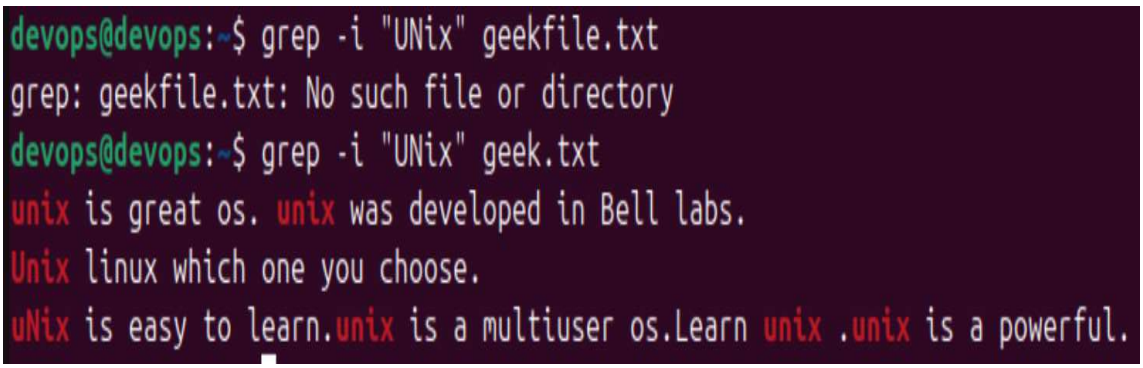
grep '[search_pattern]' [file_name]

```
cat > geek.txt
```

unix is great os. unix was developed in Bell labs.
learn operating system.
Unix linux which one you choose.
uNix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.

1. Case insensitive search

```
grep -i "UNix" geek.txt
```



2. Displaying the Count of Number of Matches Using grep

```
grep -c "unix" geek.txt
```

```
devops@devops:~$ grep -c "unix" geek.txt
2
devops@devops:~$ grep -ic "unix" geek.txt
3
```

3. Display the File Names that Matches the Pattern Using grep

```
grep -l "unix" *
```

or

```
grep -l "unix" geek.txt file1.txt f3.xt f4.txt
```

```
devops@devops:~$ grep -l "unix" *
grep: demo: Is a directory
grep: Desktop: Is a directory
grep: Documents: Is a directory
grep: Downloads: Is a directory
geek.txt
grep: Music: Is a directory
grep: Pictures: Is a directory
grep: Public: Is a directory
grep: snap: Is a directory
grep: Templates: Is a directory
grep: testcopy: Is a directory
grep: Videos: Is a directory
```

4. Checking for the Whole Words in a File Using grep

```
grep -w "unix" geek.txt
```

```
devops@devops:~$ grep -w "unix" geek.txt
unix is great os. unix was developed in Bell labs.
uNix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.
```

5. Displaying only the matched pattern Using grep

```
grep -o "unix" geek.txt
```

```
devops@devops:~$ grep -o "unix" geek.txt
unix
unix
unix
unix
unix
```

6. Show Line Number While Displaying the Output Using grep -n

```
grep -n "unix" geek.txt
```

```
devops@devops:~$ grep -n "unix" geek.txt
1:unix is great os. unix was developed in Bell labs.
4:uNix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.
```

7. Inverting the Pattern Match Using grep

```
grep -v "unix" geek.txt
```

```
devops@devops:~$ grep -v "unix" geek.txt
learn operating system.
Unix linux which one you choose.
```

8. Matching the Lines that Start with a String Using grep

```
grep "^unix" geek.txt
```

```
devops@devops:~$ grep "^unix" geek.txt
unix is great os. unix was developed in Bell labs.
```

9. Matching the Lines that End with a String Using grep

```
grep "os$" geek.txt
```

```
devops@devops:~$ grep "os" geek.txt
unix is great os. unix was developed in Bell labs.
Unix linux which one you choose.
uNix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.
devops@devops:~$ grep "os$" geek.txt
```

10.Specifies expression with -e option

Can use multiple times :

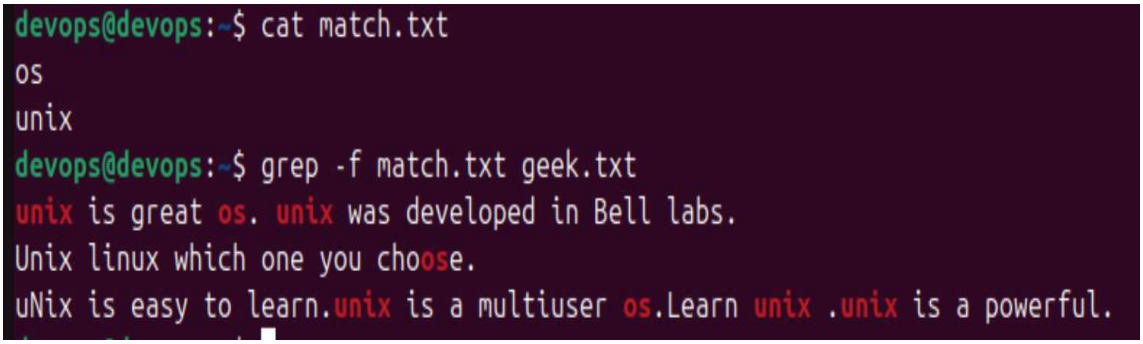
```
grep –e "unix" –e "os" geek.txt
```

```
devops@devops:~$ grep -e "unix" -e "os"  geek.txt
unix is great os. unix was developed in Bell labs.
Unix linux which one you choose.
uNix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.
devops@devops:~$ grep -e "unix"   geek.txt
unix is great os. unix was developed in Bell labs.
uNix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.
devops@devops:~$
```

11. -f file option Takes patterns from file, one per line

```
cat match.txt
```

```
grep -f match.txt geek.txt
```



12. Print n Specific Lines from a File Using grep

-A prints the searched line and n lines after the result,
-B prints the searched line and n lines before the result, and
-C prints the searched line and n lines after and before the result.

Syntax:

```
grep -A[NumberOfLines(n)] [search] [file]
grep -B[NumberOfLines(n)] [search] [file]
grep -C[NumberOfLines(n)] [search] [file]
```

```
grep -A1 learn geek.txt
```

```
devops@devops:~$ grep -A1 learn geek.txt
learn operating system.
Unix linux which one you choose.
uNix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.
devops@devops:~$ grep -A0 learn geek.txt
learn operating system.
--
uNix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.
devops@devops:~$ grep -B1 learn geek.txt
unix is great os. unix was developed in Bell labs.
learn operating system.
Unix linux which one you choose.
uNix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.
devops@devops:~$ grep -B0 learn geek.txt
learn operating system.
--
uNix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.
devops@devops:~$ grep -C1 learn geek.txt
unix is great os. unix was developed in Bell labs.
learn operating system.
Unix linux which one you choose.
uNix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.
devops@devops:~$ grep -C0 learn geek.txt
learn operating system.
```

13. Search Recursively for a Pattern in the Directory

Syntax:

```
grep -R [Search] [directory]
```

```
grep -iR yogesh /home/devops
```

```
devops@devops:/home$ grep -iR yogesh /home/devops/
grep: /home/devops/snap/firefox/6262/.config/gtk-3.0/gtk.css: No such file or directory
grep: /home/devops/snap/firefox/6262/.config/gtk-3.0/settings.ini: No such file or directory
grep: /home/devops/snap/firefox/6262/.config/gtk-2.0/gtkfilechooser.ini: No such file or director
```

Search for the word “manager”

```
grep -iR “mana[g]er” employee.txt
```

```
devops@devops:~$ grep -ir "mana[g]er" employee.txt
ajay manager account 45000
varun manager sales 50000
amit manager account 47000
```

Find Command

The find command in Linux is used to search for files and directories based on name, type, size, date, or other conditions

Syntax

```
find [path] [options] [expression]
```

1. Find files that has extension .txt in the 2. Find directories

```
find -name "*.txt" ! -empty -type f
```

Options:

Option	What It Does	Example
-name "pattern"	Searches files by name (case-sensitive).	find ~ -name "notes.txt"
-iname "pattern"	Case-insensitive name search.	find ~ -iname "notes.*"
-type f/d	Finds only files (f) or directories (d).	find /var/log -type f
-size +10M	Finds files larger than 10MB.	find / -size +100M
-mtime -7	Finds files modified in the last 7 days.	find ~ -mtime -7
-perm 644	Finds files with specific permissions.	find ~ -perm 644
-exec	Runs commands on found	find . -name

	files (e.g., delete).	"*.tmp" -exec rm {} \;
-empty	Finds empty files/directories.	find ~ -empty

1. How to find a File in Linux from the Command Line

```
find /path/to/search -options criteria
```

```
find ~ -name employee.txt
```

```
devops@devops:~$ find /home/devops/ -name employee.txt
/home/devops/employee.txt
devops@devops:~$ find -name employee.txt
./employee.txt
devops@devops:~$ find ~ -name employee.txt
/home/devops/employee.txt
```

2. How to Search Files with a Pattern Using `find` Command in Linux

```
find ./ -name test.txt
find /home/devops/ -name employee.txt
```

```
devops@devops:~$ find ./ -name test.txt
./test.txt
devops@devops:~$ find /home/devops/ -name employee.txt
/home/devops/employee.txt
```

3. Find files that has extension .txt in the

```
find -name "*.txt" ! -empty -type f
```

```
devops@devops:~$ pwd
/home/devops
devops@devops:~$ ls
demo.txt  Documents  employee.txt  geek.txt   Music      Pictures  snap       testcopy  Videos
Desktop   Downloads  file          match.txt  output.txt  Public    Templates  test.txt
devops@devops:~$ find -name "*.txt" ! -empty -type f
./output.txt
./match.txt
./geek.txt
./snap/firefox/common/.mozilla/firefox/a2jrulq1.default/serviceworker.txt
./snap/firefox/common/.mozilla/firefox/a2jrulq1.default/pkcs11.txt
./employee.txt
./testcopy/hello.txt
./testcopy/file1.txt
./.cache/tracker3/files/last-crawl.txt
./.cache/tracker3/files/first-index.txt
./test.txt
devops@devops:~$ cat ./output.txt
ajay manager
sunil clerk
varun manager
amit manager
tarun peon
deepak clerk
sunil peon
satvik director
devops@devops:~$
```

4. Find directories

```
find -name "*" ! -empty -type d
```

```
devops@devops:~$ find -name "*" ! -empty -type d
```

5. Search Files with Specific Permissions Using `find` Command in Linux

```
find /home/devops -perm 664
```

```
devops@devops:~$ find /home/devops/ -perm 666
/home/devops/test.txt
```

6. Display Repository Hierarchy Using `find` Command in Linux

```
find /home/devops -type d
```

```
devops@devops:~$ find /home/devops/ -type d
/home/devops/
/home/devops/Videos
/home/devops/.ssh
/home/devops/Downloads
/home/devops/snap
/home/devops/snap/firefox
/home/devops/snap/firefox/6262
```

7. Search Text Within Multiple Files Using `find` Command in Linux

```
find ./ -type f -name "*.txt" -exec grep 'Geek'  {} \;
```

```
devops@devops:~$ find /home/devops/ -type f -name "*.txt" -exec grep "Apple" {} \;
Apple Apple
```

8. Find Files by When They Were Modified Using `find` Command in Linux

```
devops@devops:~$ find /home/devops/ -mtime -7
```

9. Difficult: Find .txt files that are not empty and search for exact word “employee” with
their file names.

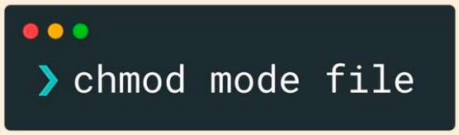
CHMOD

The chmod command on Linux is used to manage permissions for files and directories.

chmod

To change the permissions of a file or directory, we can use the chmod command (change mode).

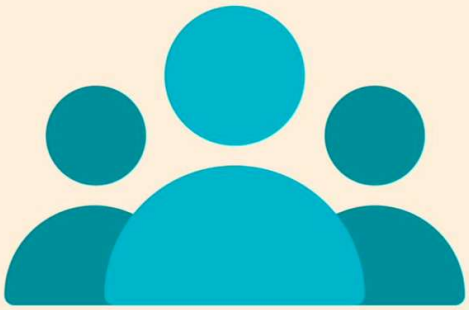
To use chmod to alter permissions, we need to tell it:
- **Who** we are changing permissions for
- **What** change are we making? Adding? Removing?
- **Which** permissions are we setting?



Multiple Users

Unix and unix-like systems are **multiuser** operating systems. More than one person can be using the same computer at the same time (though this is tough logistically with only one display and keyboard!)

Way back when, computers were crazy expensive, massive machines. A university might only have one computer, but dozens of terminals sprinkled across campus.



chmod

When specifying permissions with chmod, we use a special syntax to write permission statements.

```
❯ chmod mode file
```

First, we specify the "who" using the following values:

- **u** - user (the owner of the file)
- **g** - group (members of the group the file belongs to
- **o** - others (the "world"
- **a** - all of the above

All Togther Now

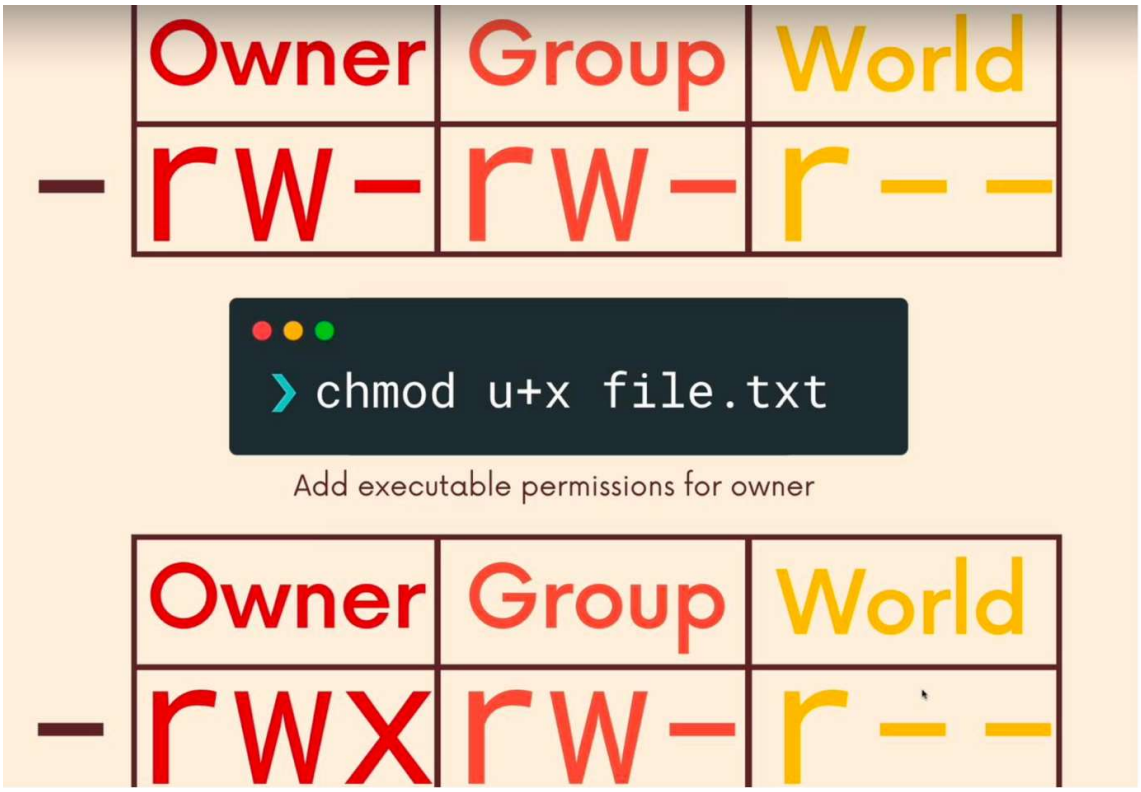
Next, we tell chmod "what" we are doing using the following characters:

- - (minus sign) removes the permission
- + (plus sign) grants the permission
- = (equals sign) set a permission and removes others

```
❯ chmod mode file
```

Finally, the "which" values are:

- r - the read permission
- w - the write permission
- x - the execute permission



1. Create a file name happy.txt.

```
touch happy.txt
```

```
devops@devops:~$ touch happy.txt
devops@devops:~$ ls -l happy.txt
-rw-rw-r-- 1 devops devops 0 Jun 13 06:46 happy.txt
```

2. Change its default permissions.

```
chmod u+x happy.txt
```

```
devops@devops:~$ ls -l happy.txt
-rw-rw-r-- 1 devops devops 0 Jun 13 06:46 happy.txt
devops@devops:~$ chmod u+x happy.txt
devops@devops:~$ ls -l happy.txt
-rwxrw-r-- 1 devops devops 0 Jun 13 06:46 happy.txt
```

3. Write permission for all.

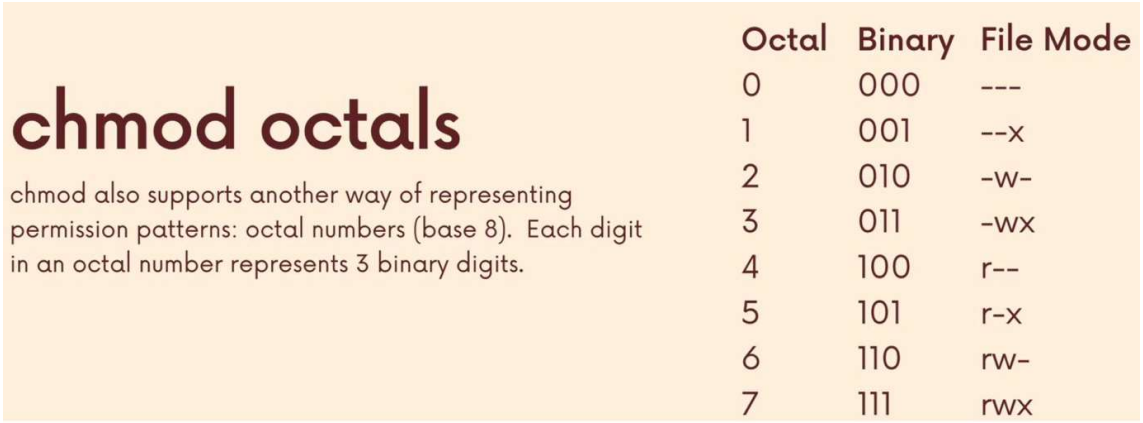
```
chmod a+w happy.txt
```

```
devops@devops:~$ touch happy.txt
devops@devops:~$ ls -l happy.txt
-rw-rw-r-- 1 devops devops 0 Jun 13 06:46 happy.txt
devops@devops:~$ chmod u+x happy.txt
devops@devops:~$ ls -l happy.txt
-rwxrw-r-- 1 devops devops 0 Jun 13 06:46 happy.txt
devops@devops:~$ chmod a+w happy.txt
devops@devops:~$ ls -l happy.txt
-rwxrw-rw- 1 devops devops 0 Jun 13 06:46 happy.txt
```

4. Add write and Read permission to the user.

```
chmod u+wx happy.txt
```

```
devops@devops:~$ touch happy.txt
devops@devops:~$ ls -l happy.txt
-rw-rw-r-- 1 devops devops 0 Jun 13 06:46 happy.txt
devops@devops:~$ chmod u+x happy.txt
devops@devops:~$ ls -l happy.txt
-rwxrw-r-- 1 devops devops 0 Jun 13 06:46 happy.txt
devops@devops:~$ chmod a+w happy.txt
devops@devops:~$ ls -l happy.txt
-rwxrw-rw- 1 devops devops 0 Jun 13 06:46 happy.txt
devops@devops:~$ chmod a+wx happy.txt
devops@devops:~$ ls -l happy.txt
-rwxrwxrwx 1 devops devops 0 Jun 13 06:46 happy.txt
```

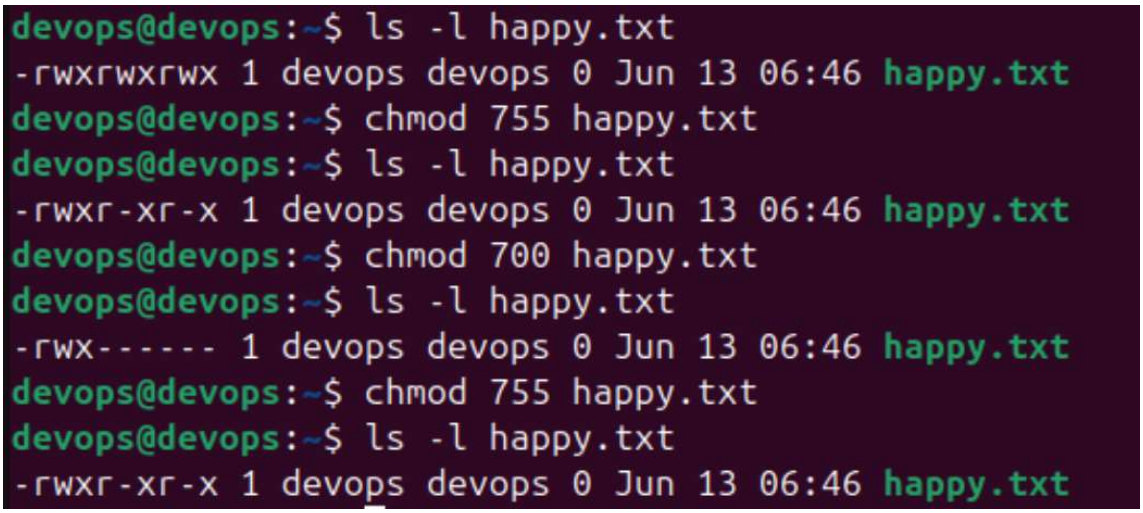



```
chmod 755 happy.txt
```

```
devops@devops:~$ ls -l happy.txt
-rwxrwxrwx 1 devops devops 0 Jun 13 06:46 happy.txt
devops@devops:~$ chmod 755 happy.txt
devops@devops:~$ ls -l happy.txt
-rwxr-xr-x 1 devops devops 0 Jun 13 06:46 happy.txt
```

5. Practice: change the permission of happy.txt to give all permission to user(read,

write, execute) and no permission to anyone else.



Environment variables

1. To print all environment variables

```
printenv | less
```

2. Search USER variable

```
printenv | grep USER
```

3.echo $USER

```
echo $USER
```

4. echo $HOME

```
echo $HOME
```

5. echo $PWD

```
echo $PWD
```

```
devops@devops:~$ printenv | grep USER
USERNAME=devops
USER=devops
devops@devops:~$ echo $USER
devops
devops@devops:~$ echo $HOME
/home/devops
devops@devops:~$ echo $PWD
/home/devops
```

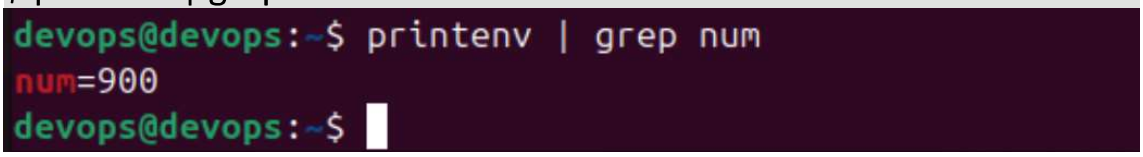
Shell variables

```
num=500
echo $num
```

```
devops@devops:~$ num=100
devops@devops:~$ echo $num
100
```

These exists only in the shell, not global env. Variables
To create global variables

```
export num=900
$ printenv | grep num
```

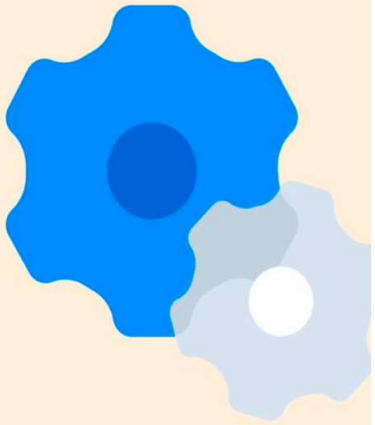


Start up Files

Startup Files

When we log in, the shell reads information from startup files. First, the shell reads from global config files that effect the environment for all users. Then, the shell reads startup files for specific users.

The specific files the shell reads from depends on the type of session: login vs. non-login shell sessions



For login sessions:

- /etc/profile - global config for all users
- ~/.bash_profile - user's personal config file
- ~/.bash_login - read if bash_profile isn't found
- ~/.profile - used if previous two aren't found