

Homework 8

Student Name: Anoosha Sagar

Student ID: 605028604

1. UNIX Tutorial

UNIX

- UNIX is an operating system which was first developed in the 1960s, and has been in development ever since.
- The UNIX operating system is made up of three parts: the kernel, the shell, and the programs.
- Versions of UNIX: Sun Solaris, GNU/Linux, and MacOS
- Everything in UNIX is either a file or process.
- All files are grouped together in a grouped directory structure. The top of the hierarchy is traditionally called *root*.

UNIX Commands

mkdir and ls

mkdir

- Used for creating a directory

ls

- It lists the contents of the current working directory
- Example usage:
 - **ls**: list files and directories whose names do not start with a dot (.)
 - **ls -a**: list all files in the directory
 - **ls -lrt**: long listing of files, sorted by modification time in reverse

```
anoosha@DESKTOP-E89P26T:~$ mkdir sampleDir
anoosha@DESKTOP-E89P26T:~$ ls
readme.txt  sampleDir  stats201
anoosha@DESKTOP-E89P26T:~$ ls -a
.  ..  .bash_logout  .bashrc  .profile  readme.txt  sampleDir  stats201
anoosha@DESKTOP-E89P26T:~$ ls -lrt
total 0
drwxrwxrwx 0 anoosha anoosha 4096 Nov 25 21:24 stats201
-rw-rw-rw- 1 anoosha anoosha   0 Nov 27 18:54 readme.txt
drwxrwxrwx 0 anoosha anoosha 4096 Nov 27 18:57 sampleDir
anoosha@DESKTOP-E89P26T:~$
```

cd

- The **cd** command is used to change the directory
- Examples:
 - **cd <<directory name>>**: change the directory to the directory listed in the command
 - **cd .**: stay where you are (dot(.) denotes the current directory)
 - **cd ..**: move to the parent directory (..) denotes parent of the current directory)

```
anoosha@DESKTOP-E89P26T:~$ cd stats201
anoosha@DESKTOP-E89P26T:~/stats201$ ls -a
.  ..
anoosha@DESKTOP-E89P26T:~/stats201$ cd .
anoosha@DESKTOP-E89P26T:~/stats201$ cd ..
anoosha@DESKTOP-E89P26T:~$ cd unixstuff/
```

ls Using Pathnames and pwd

pwd

- displays the path of the current directory

Examples of ls using pathnames

- **ls <<directory>>**: list contents of directory specified
- **ls <<directory>> -a**: list all contents of the directory specified

```
anoosha@DESKTOP-E89P26T:~$ ls unixstuff/
backups
anoosha@DESKTOP-E89P26T:~$ ls unixstuff/backups/ -a
.  ..
anoosha@DESKTOP-E89P26T:~$ pwd
/home/anoosha
anoosha@DESKTOP-E89P26T:~$
```

cp, mv, rm, rmdir

cp

- **cp file1 file2**: copy file1 in the current working directory and call it file2
- Examples:
 - **cp filename .**: copy filename to the current directory keeping the name same

mv

- **mv file1 file2** moves (or renames) file1 to file2
- Examples:
 - **mv file directory/.**: move file to the specified directory and keep the name same

rm

- **rm file** deletes the specified file

rmdir

- **rmdir directory** deletes the specified directory

```

cp: /home/anoosha/readme.txt and ./readme.txt are the same file
anoosha@DESKTOP-E89P26T:~$ cd ~/unixstuff/
anoosha@DESKTOP-E89P26T:~/unixstuff$ cp ~/readme.txt .
anoosha@DESKTOP-E89P26T:~/unixstuff$ ls -a
.  ..  backups  readme.txt
anoosha@DESKTOP-E89P26T:~/unixstuff$ cp ~/readme.txt readme.bak
anoosha@DESKTOP-E89P26T:~/unixstuff$ ls -a
.  ..  backups  readme.bak  readme.txt
anoosha@DESKTOP-E89P26T:~/unixstuff$ mv readme.bak backups/.
anoosha@DESKTOP-E89P26T:~/unixstuff$ ls backups
readme.bak
anoosha@DESKTOP-E89P26T:~/unixstuff$ cp readme.txt tempfile.txt
anoosha@DESKTOP-E89P26T:~/unixstuff$ ls
backups  readme.txt  tempfile.txt
anoosha@DESKTOP-E89P26T:~/unixstuff$ rm tempfile.txt
anoosha@DESKTOP-E89P26T:~/unixstuff$ ls
backups  readme.txt
anoosha@DESKTOP-E89P26T:~/unixstuff$ mkdir tempDir
anoosha@DESKTOP-E89P26T:~/unixstuff$ ls
backups  readme.txt  tempDir
anoosha@DESKTOP-E89P26T:~/unixstuff$ rmdir tempDir/
anoosha@DESKTOP-E89P26T:~/unixstuff$ ls
backups  readme.txt
anoosha@DESKTOP-E89P26T:~/unixstuff$

```

cat, less, head, and tail

cat

- **cat filename** displays the contents of the file on the terminal screen

less

- **less filename** writes the contents of the file onto a separate screen one page at a time

head

- **head filename** writes the first 10 lines of the file to the terminal screen
- **head -n filename** writes the first n lines of the file to the terminal screen

tail

- **tail filename** writes the last 10 lines of the file to the terminal screen
- **tail -n filename** writes the last n lines of the file to the terminal screen

```
anoosha@DESKTOP-E89P26T: ~/unixstuff
anoosha@DESKTOP-E89P26T:~/unixstuff$ cat readme.txt
Hello....Whats up!!!
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
anoosha@DESKTOP-E89P26T:~/unixstuff$ less readme.txt
anoosha@DESKTOP-E89P26T:~/unixstuff$ head readme.txt
Hello....Whats up!!!
1
2
3
4
5
6
7
8
9
anoosha@DESKTOP-E89P26T:~/unixstuff$ head -2 readme.txt
Hello....Whats up!!!
1
anoosha@DESKTOP-E89P26T:~/unixstuff$ tail -5 readme.txt
12
13
14
15
anoosha@DESKTOP-E89P26T:~/unixstuff$
```

Simple searching using less

Steps

1. less filename
2. /[word to search]

less finds and highlights the keywords


```
anoosha@DESKTOP-E89P26T: ~/unixstuff
anoosha@DESKTOP-E89P26T:~/unixstuff$ grep up readme.txt
Hello...Whats up!!!
anoosha@DESKTOP-E89P26T:~/unixstuff$ grep Up readme.txt
anoosha@DESKTOP-E89P26T:~/unixstuff$ grep -i Up readme.txt
Hello...Whats up!!!
anoosha@DESKTOP-E89P26T:~/unixstuff$ grep -i 'whats Up!!!' readme.txt
Hello...Whats up!!!
anoosha@DESKTOP-E89P26T:~/unixstuff$ wc -w readme.txt
17 readme.txt
anoosha@DESKTOP-E89P26T:~/unixstuff$ wc -l readme.txt
17 readme.txt
anoosha@DESKTOP-E89P26T:~/unixstuff$ _
```

Redirecting the Output

Redirect output of command

Example: cat > filename

Appending to a file

Example: cat >> filename

Concatenate files

Example: cat file1 file2 > file3

```
anoosha@DESKTOP-E89P26T: ~
anoosha@DESKTOP-E89P26T:~/unixstuff$ cd .
anoosha@DESKTOP-E89P26T:~/unixstuff$ cd ..
anoosha@DESKTOP-E89P26T:~$ cat
This is Stats 201
This is Stats 201
The course name is Statistics Programming
The course name is Statistics Programming
anoosha@DESKTOP-E89P26T:~$ cat > sampleFile
Stats201
Statistics Programming
Tuesday Thursday 11-12.15
anoosha@DESKTOP-E89P26T:~$ ls
readme.txt  sampleFile  unixstuff
anoosha@DESKTOP-E89P26T:~$ cat sampleFile
Stats201
Statistics Programming
Tuesday Thursday 11-12.15
anoosha@DESKTOP-E89P26T:~$ cat >> sampleFile
Haines Hall
Discussion: Thursday
anoosha@DESKTOP-E89P26T:~$ cat sampleFile
Stats201
Statistics Programming
Tuesday Thursday 11-12.15
Haines Hall
Discussion: Thursday
anoosha@DESKTOP-E89P26T:~$ cat > sampleFile2
Other Courses: Advanced Computer Networks, Cryptography
anoosha@DESKTOP-E89P26T:~$ cat sampleFile sampleFile2 > final
anoosha@DESKTOP-E89P26T:~$ cat final
Stats201
Statistics Programming
Tuesday Thursday 11-12.15
Haines Hall
Discussion: Thursday
Other Courses: Advanced Computer Networks, Cryptography
anoosha@DESKTOP-E89P26T:~$
```

Redirecting the input

Example: sort < filename

Using < we can redirect the input to come from a file rather than keyboard

Example: sort <f file1 > file2

Outputs sorted file to another file

```
anoosha@DESKTOP-E89P26T:~$  
anoosha@DESKTOP-E89P26T:~$ sort  
dog  
cat  
bird  
ape  
ape  
bird  
cat  
dog  
anoosha@DESKTOP-E89P26T:~$ sort < final  
Discussion: Thursday  
Haines Hall  
Other Courses: Advanced Computer Networks, Cryptography  
Statistics Programming  
Stats201  
Tuesday Thursday 11-12.15  
anoosha@DESKTOP-E89P26T:~$ sort < final > final1  
anoosha@DESKTOP-E89P26T:~$ cat final1  
Discussion: Thursday  
Haines Hall  
Other Courses: Advanced Computer Networks, Cryptography  
Statistics Programming  
Stats201  
Tuesday Thursday 11-12.15  
anoosha@DESKTOP-E89P26T:~$
```

Pipes

Used when we want to connect the output of a command to the input of a second command. The symbol for pipe is a vertical bar (|)

```
anoosha@DESKTOP-E89P26T:~$ who  
anoosha@DESKTOP-E89P26T:~$ who > names.txt  
anoosha@DESKTOP-E89P26T:~$ sort < names.txt  
anoosha@DESKTOP-E89P26T:~$ who | sort  
anoosha@DESKTOP-E89P26T:~$ who | wc -l  
0
```

Wildcards

*

- The character * is called a wildcard, and will match against none or more character(s) in a file (or directory).
- Example:
 - **ls list***: list all files in the current directory starting with list...
 - **ls *list**: list all files in the current directory ending with ...list

?

- The ? character will match exactly one character
- Example
 - **ls ?list**


```
Select anoosha@DESKTOP-E89P26T: ~
anoosha@DESKTOP-E89P26T:~$ ls
final final1 names.txt readme.txt sampleDir sampleFile sampleFile2 stats201 unixstuff
anoosha@DESKTOP-E89P26T:~$ ls sample*
sampleFile sampleFile2

sampleDir:
anoosha@DESKTOP-E89P26T:~$ ls *File
sampleFile
anoosha@DESKTOP-E89P26T:~$ ls *File*
sampleFile sampleFile2
anoosha@DESKTOP-E89P26T:~$ ls ?ames.txt
names.txt
anoosha@DESKTOP-E89P26T:~$
```

Online Manual

Example: man ls

Loads the manual page for the ls command

```
anoosha@DESKTOP-E89P26T: ~
LS(1) User Commands LS(1)
NAME
  ls - list directory contents

SYNOPSIS
  ls [OPTION]... [FILE]...

DESCRIPTION
  List information about the FILES (the current directory by default). Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

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

  -a, --all
      do not ignore entries starting with .

  -A, --almost-all
      do not list implied . and ..

  --author
      with -l, print the author of each file

  -b, --escape
      print C-style escapes for nongraphic characters

  --block-size=SIZE
      scale sizes by SIZE before printing them; e.g., '--block-size=M' prints sizes in units of 1,048,576 bytes; see SIZE format below

  -B, --ignore-backups
      do not list implied entries ending with ~

  -c
      with -lt: sort by, and show, ctime (time of last modification of file status information); with -l: show ctime and sort by name; otherwise: sort by ctime, newest first

  -C
      list entries by columns

  --color[=WHEN]
      colorize the output; WHEN can be 'always' (default if omitted), 'auto', or 'never'; more info below

  -d, --directory
      list directories themselves, not their contents

  -D, --dired
      generate output designed for Emacs' dired mode

Manual page ls(1) line 1 (press h for help or q to quit)
what is
```

Example: *whatis wc*

Gives a one line description of the command

```
anoosha@DESKTOP-E89P26T:~$ whatis wc
wc (1) - print newline, word, and byte counts for each file
anoosha@DESKTOP-E89P26T:~$
```

Apropos

apropos keyword will give the commands with the keyword in their manual page header.

```
anoosha@DESKTOP-E89P26T:~$ apropos copy
cp (1) - copy files and directories
cpgr (8) - copy with locking the given file to the password or group file
cpio (1) - copy files to and from archives
cppw (8) - copy with locking the given file to the password or group file
dd (1) - convert and copy a file
debconf-copydb (1) - copy a debconf database
git-checkout-index (1) - Copy files from the index to the working tree
install (1) - copy files and set attributes
ntfscp (8) - copy file to an NTFS volume.
rcp (1) - secure copy (remote file copy program)
rsync (1) - a fast, versatile, remote (and local) file-copying tool
scp (1) - secure copy (remote file copy program)
ssh-copy-id (1) - use locally available keys to authorise logins on a remote machine
xfs_copy (8) - copy the contents of an XFS filesystem
xfs_metadump (8) - copy XFS filesystem metadata to a file
xfs_rtcp (8) - XFS realtime copy command
anoosha@DESKTOP-E89P26T:~$
```

chmod

- This command is used for changing the access rights of a file
- Only the owner of a file can use chmod to change the permissions of a file
- Example of access rights for a file

rw-rw-rw-	a file that everyone can read, write and execute (and delete).
-rw-----	a file that only the owner can read and write - no-one else can read or write and no-one has execution rights (e.g. your mailbox file).

- Options of chmod

Symbol	Meaning
u	user
g	group
o	other
a	all
r	read

w	write (and delete)
x	execute (and access directory)
+	add permission
-	take away permission

- Example usage
 - **chmod go-rwx final1** : remove read write and execute permissions for the file called final1 for group and others

```

anoosha@DESKTOP-E89P26T:~$ ls -lrt
total 0
drwxrwxrwx 0 anoosha anoosha 4096 Nov 25 21:24 stats201
drwxrwxrwx 0 anoosha anoosha 4096 Nov 27 18:57 sampleDir
-rw-rw-rw- 1 anoosha anoosha 22 Nov 27 19:07 readme.txt
drwxrwxrwx 0 anoosha anoosha 4096 Nov 27 19:14 unixstuff
-rw-rw-rw- 1 anoosha anoosha 91 Nov 27 22:44 sampleFile
-rw-rw-rw- 1 anoosha anoosha 56 Nov 27 22:45 sampleFile2
-rw-rw-rw- 1 anoosha anoosha 147 Nov 27 22:46 final
-rw-rw-rw- 1 anoosha anoosha 147 Nov 27 22:48 final1
-rw-rw-rw- 1 anoosha anoosha 0 Nov 27 22:49 names.txt
anoosha@DESKTOP-E89P26T:~$ chmod go-rwx final1
anoosha@DESKTOP-E89P26T:~$ ls -lrt
total 0
drwxrwxrwx 0 anoosha anoosha 4096 Nov 25 21:24 stats201
drwxrwxrwx 0 anoosha anoosha 4096 Nov 27 18:57 sampleDir
-rw-rw-rw- 1 anoosha anoosha 22 Nov 27 19:07 readme.txt
drwxrwxrwx 0 anoosha anoosha 4096 Nov 27 19:14 unixstuff
-rw-rw-rw- 1 anoosha anoosha 91 Nov 27 22:44 sampleFile
-rw-rw-rw- 1 anoosha anoosha 56 Nov 27 22:45 sampleFile2
-rw-rw-rw- 1 anoosha anoosha 147 Nov 27 22:46 final
-rw----- 1 anoosha anoosha 147 Nov 27 22:48 final1
-rw-rw-rw- 1 anoosha anoosha 0 Nov 27 22:49 names.txt
anoosha@DESKTOP-E89P26T:~$

```

Processes and Jobs

ps

- Used to see information about processes, with their associated PID and status
- Use **&** to run a process in background – e.g., **sleep 10 &**
- To put a current foreground process in the background, use **bg**

jobs

- It lists all running, backgrounded, and suspended processes
- To restart (foreground) a suspended process, use **fg** – e.g., **fg %jobnumber**

kill

- Used for terminating processes
- Usage – **kill %jobnumber** or **kill PID**

FW FW FW 1 anoosha anoosha 0 Nov 27 22:49 names: CXC

```
anoosha@DESKTOP-E89P26T:~$ ps
  PID TTY          TIME CMD
    2 tty1      00:00:01 bash
   275 tty1      00:00:00 ps
anoosha@DESKTOP-E89P26T:~$ sleep 10
anoosha@DESKTOP-E89P26T:~$ sleep 10 &
[1] 277
anoosha@DESKTOP-E89P26T:~$ sleep 1000
^Z[1]  Done                  sleep 10

[2]+  Stopped                  sleep 1000
anoosha@DESKTOP-E89P26T:~$ bg
[2]+ sleep 1000 &
anoosha@DESKTOP-E89P26T:~$ jobs
[2]+  Running                  sleep 1000 &
anoosha@DESKTOP-E89P26T:~$ fg %1
-bash: fg: %1: no such job
anoosha@DESKTOP-E89P26T:~$ fg %2
sleep 1000
^Z^X
[2]+  Stopped                  sleep 1000
anoosha@DESKTOP-E89P26T:~$ s
s: command not found
anoosha@DESKTOP-E89P26T:~$ ps
  PID TTY          TIME CMD
    2 tty1      00:00:01 bash
   278 tty1      00:00:00 sleep
   281 tty1      00:00:00 ps
anoosha@DESKTOP-E89P26T:~$ jobs
[2]+  Stopped                  sleep 1000
anoosha@DESKTOP-E89P26T:~$
```

```
anoosha@DESKTOP-E89P26T: ~
anoosha@DESKTOP-E89P26T:~$ sleep 100 &
[3] 283
anoosha@DESKTOP-E89P26T:~$ kill %283
-bash: kill: %283: no such job
anoosha@DESKTOP-E89P26T:~$ sleep 100 &
[4] 284
anoosha@DESKTOP-E89P26T:~$ kill %4
[4]- Terminated              sleep 100
anoosha@DESKTOP-E89P26T:~$ sleep 1000 &
[4] 285
anoosha@DESKTOP-E89P26T:~$ ps
  PID TTY          TIME CMD
    2 tty1      00:00:01 bash
   278 tty1      00:00:00 sleep
   283 tty1      00:00:00 sleep
   285 tty1      00:00:00 sleep
   286 tty1      00:00:00 ps
anoosha@DESKTOP-E89P26T:~$ kill 285
[4]- Terminated              sleep 1000
anoosha@DESKTOP-E89P26T:~$ ps
  PID TTY          TIME CMD
    2 tty1      00:00:01 bash
   278 tty1      00:00:00 sleep
   283 tty1      00:00:00 sleep
   287 tty1      00:00:00 ps
anoosha@DESKTOP-E89P26T:~$
```

df

It reports the space left on the system

du

It outputs the number of kilobytes used by each subdirectory

du -s * - will show a summary of all files and directories

```
anoosha@DESKTOP-E89P26T: ~  
anoosha@DESKTOP-E89P26T:~$ df  
Filesystem      1K-blocks    Used Available Use% Mounted on  
rootfs          534966268 73809512 461156756 14% /  
none            534966268 73809512 461156756 14% /dev  
none            534966268 73809512 461156756 14% /run  
none            534966268 73809512 461156756 14% /run/lock  
none            534966268 73809512 461156756 14% /run/shm  
none            534966268 73809512 461156756 14% /run/user  
C:              534966268 73809512 461156756 14% /mnt/c  
D:              13418492 11794044 1624448 88% /mnt/d  
E:              427082748 341480 426741268 1% /mnt/e  
anoosha@DESKTOP-E89P26T:~$  
anoosha@DESKTOP-E89P26T:~$  
anoosha@DESKTOP-E89P26T:~$ du -s *  
0      final  
0      final1  
0      names.txt  
0      readme.txt  
0      sampleDir  
0      sampleFile  
0      sampleFile2  
0      stats201  
0      unixstuff
```

gzip

This command reduces the size of the file (compress the file)

zcat

This command reads the gzipped files without uncompressing them first

```
anoosha@DESKTOP-E89P26T: ~
-rw-rw-rw- 1 anoosha anoosha 731 Nov 27 23:08 abc.txt
-rw-rw-rw- 1 anoosha anoosha 147 Nov 27 22:46 final
-rw-rw-rw- 1 anoosha anoosha 150 Nov 27 22:48 final1.gz
-rw-rw-rw- 1 anoosha anoosha 0 Nov 27 22:49 names.txt
-rw-rw-rw- 1 anoosha anoosha 22 Nov 27 19:07 readme.txt
drwxrwxrwx 0 anoosha anoosha 4096 Nov 27 18:57 sampleDir
-rw-rw-rw- 1 anoosha anoosha 91 Nov 27 22:44 sampleFile
-rw-rw-rw- 1 anoosha anoosha 56 Nov 27 22:45 sampleFile2
drwxrwxrwx 0 anoosha anoosha 4096 Nov 25 21:24 stats201
drwxrwxrwx 0 anoosha anoosha 4096 Nov 27 19:14 unixstuff
anoosha@DESKTOP-E89P26T:~$ gzip abc.txt
anoosha@DESKTOP-E89P26T:~$ ls -l
total 0
-rw-rw-rw- 1 anoosha anoosha 118 Nov 27 23:08 abc.txt.gz
-rw-rw-rw- 1 anoosha anoosha 147 Nov 27 22:46 final
-rw-rw-rw- 1 anoosha anoosha 150 Nov 27 22:48 final1.gz
-rw-rw-rw- 1 anoosha anoosha 0 Nov 27 22:49 names.txt
-rw-rw-rw- 1 anoosha anoosha 22 Nov 27 19:07 readme.txt
drwxrwxrwx 0 anoosha anoosha 4096 Nov 27 18:57 sampleDir
-rw-rw-rw- 1 anoosha anoosha 91 Nov 27 22:44 sampleFile
-rw-rw-rw- 1 anoosha anoosha 56 Nov 27 22:45 sampleFile2
drwxrwxrwx 0 anoosha anoosha 4096 Nov 25 21:24 stats201
drwxrwxrwx 0 anoosha anoosha 4096 Nov 27 19:14 unixstuff
anoosha@DESKTOP-E89P26T:~$ zcat abc.txt.gz
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
wwwwww
ddddddddddddddddddddddddddddddddd
cccccccccccccccccccccccccccccccccc
ddddddddddddddddddddddddddddddddd
wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww
ffffffffffffffffffffffffffffffffffff
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee
ffffffffffffffffffffffffffffffffffffv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
44444444444444444444444444444444
6666666666666666666666666666666666
fggggggggggggrefvbyh
3333333333333333333333333333333333
ffffffffffffffffffffffffffffffffffff

vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
```

file

This command classifies the named files according to the type of data they contain.

diff

This command compares the contents of two files and displays the differences

find

This searches through the directories for files and directories with a given name, date, size, or any other attribute specified.

```

anoosha@DESKTOP-E89P26T:~$ file *
abc.txt.gz:  gzip compressed data, was "abc.txt", last modified: Tue Nov 28 07:08:42 2017, from Unix
final:      ASCII text
final1.gz:  gzip compressed data, was "final1", last modified: Tue Nov 28 06:48:00 2017, from Unix
names.txt:  empty
readme.txt: ASCII text
sampleDir:  directory
sampleFile: ASCII text
sampleFile2: ASCII text
stats201:   directory
unixstuff:  directory
anoosha@DESKTOP-E89P26T:~$ diff sampleFile sampleFile2
1,5c1
< Stats201
< Statistics Programming
< Tuesday Thursday 11-12.15
< Haines Hall
< Discussion: Thursday
---
> Other Courses: Advanced Computer Networks, Cryptography
anoosha@DESKTOP-E89P26T:~$ find . -name "*.txt" -print
./names.txt
./readme.txt
./unixstuff/readme.txt

```

history

This command shows the command history list

```

anoosha@DESKTOP-E89P26T: ~
./unixstuff/readme.txt
anoosha@DESKTOP-E89P26T:~$ find . -size +1M -ls
anoosha@DESKTOP-E89P26T:~$ history
 1  ls
 2  touch readme.txt
 3  ls
 4  cat readme.txt
 5  cls
 6  clr
 7  clear
 8  ls -lrt
 9  ls
10  ls -a
11  ls -lrt
12  mkdir sampleDir
13  ls
14  ls -a
15  ls -lrt
16  mkdir unixstuff

```

2. Git Tutorial

GitHub

GitHub is a Web-based Git version control repository hosting service. It offers all of the distributed version control and source code management (SCM) functionality of Git as well as adding its own features.

It provides access control and several collaboration features such as bug tracking, feature requests, task management, and wikis for every project.

My GitHub Profile

The screenshot shows the GitHub 'Public profile' page. On the left is a sidebar with navigation links: Personal settings, Profile, Account, Emails, Notifications, Billing, SSH and GPG keys, Security, Blocked users, Repositories, Organizations, Saved replies, Applications, and Developer settings. The main content area is titled 'Public profile' and contains several input fields: Name (filled with 'Anoosha Sagar'), Public email (a dropdown menu with 'Select a verified email to display'), Bio (a text area with the placeholder 'Tell us a little bit about yourself'), URL, Company, and Location. To the right of the 'Public email' field is a note: 'You have set your email address to private. To toggle email privacy, go to [email settings](#) and uncheck "Keep my email address private."'. Below the 'Bio' field is another note: 'You can @mention other users and organizations to link to them.' Below the 'Company' field is a note: 'You can @mention your company's GitHub organization to link it.' On the right side of the profile section is a 'Profile picture' section with a photo of a woman and an 'Upload new picture' button.

Git Setup

1. Get a GitHub account
2. Download and install git
3. Set up git with your username and email

```
anoos@DESKTOP-E89P26T MINGW64 ~  
$ git config --global user.name "asagar06"  
  
anoos@DESKTOP-E89P26T MINGW64 ~  
$ git config --global user.email "anooshasagar@gmail.com"  
  
anoos@DESKTOP-E89P26T MINGW64 ~  
$ git config --global color.ui true  
  
anoos@DESKTOP-E89P26T MINGW64 ~  
$ git config --global core.editor emacs  
  
anoos@DESKTOP-E89P26T MINGW64 ~  
$
```

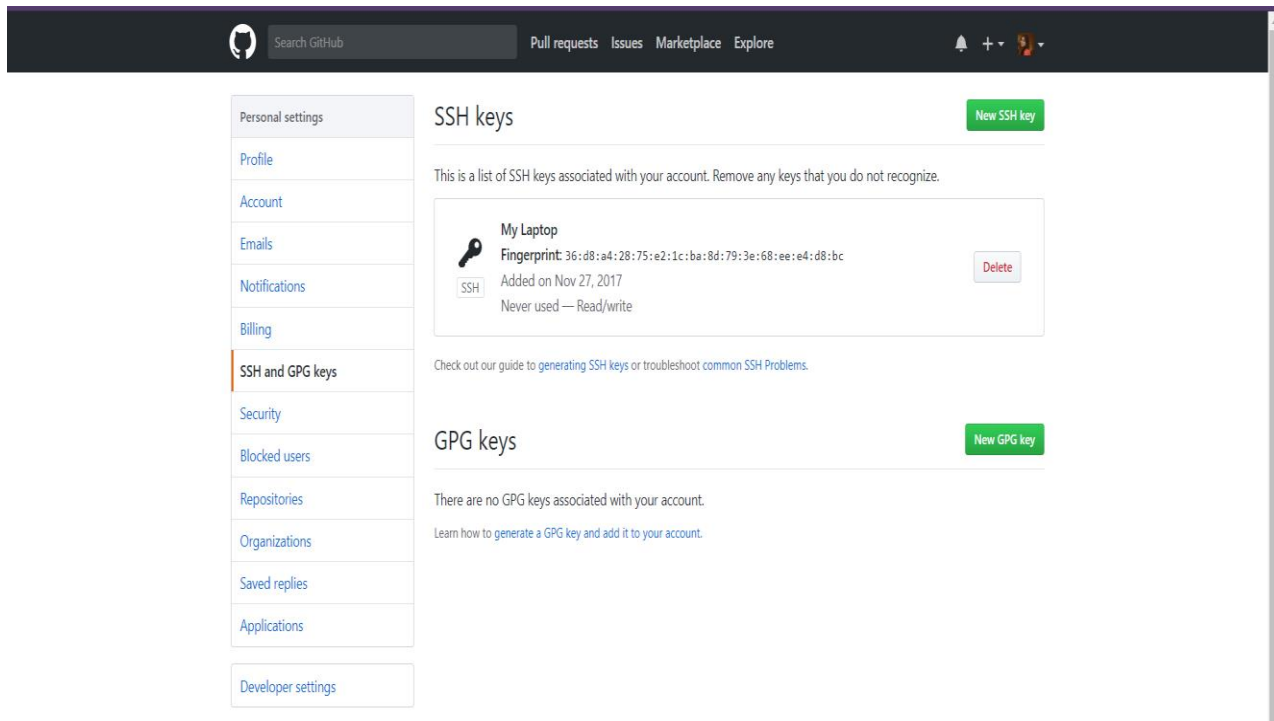

4. Set up ssh on the computer

```
anoos@DESKTOP-E89P26T MINGW64 ~
$ ssh-keygen -t rsa -C "anooshasagar@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/anoos/.ssh/id_rsa):
Created directory '/c/Users/anoos/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/anoos/.ssh/id_rsa.
Your public key has been saved in /c/Users/anoos/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:gPhocZvevTepIPfyjKIfwFRIFuWkR5xfd1Swud3/984 anooshasagar@gmail.com
The key's randomart image is:
+---[RSA 2048]-----+
| .o+. . .oo |
| . Bo o .oo |
| o B +. o . o |
| X * . . . o |
| o B . S . |
| . . o . . |
| + + . . . |
| .+. = .+ .o |
| .o...o*o . .E|
+---[SHA256]-----+

anoos@DESKTOP-E89P26T MINGW64 ~
$ ssh -T git@github.com
The authenticity of host 'github.com (192.30.255.113)' can't be established.
RSA key fingerprint is SHA256:nThbg6kXUpJWG17E1IGOCspRomTxdCARLviKw6E5SY8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'github.com,192.30.255.113' (RSA) to the list of known hosts.
Hi asagar06! You've successfully authenticated, but GitHub does not provide shell access.

anoos@DESKTOP-E89P26T MINGW64 ~
$ |
```

5. Copy paste the ssh public key into your github account settings



The screenshot shows the GitHub web interface. On the left is a sidebar with navigation links: Personal settings, Profile, Account, Emails, Notifications, Billing, SSH and GPG keys (highlighted), Security, Blocked users, Repositories, Organizations, Saved replies, Applications, and Developer settings. The main content area is titled 'SSH keys' and includes a 'New SSH key' button. Below the title is a message: 'This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.' There is one key listed: 'My Laptop' with a key icon, fingerprint '36:d8:a4:28:75:e2:1c:ba:8d:79:3e:68:ee:e4:d8:bc', added on Nov 27, 2017, and permissions 'Never used — Read/write'. A 'Delete' button is next to it. Below the key list is a link to a guide: 'Check out our guide to [generating SSH keys](#) or [troubleshoot common SSH Problems](#).' Underneath is a section for 'GPG keys' with a 'New GPG key' button. It states: 'There are no GPG keys associated with your account.' and provides a link: 'Learn how to generate a GPG key and add it to your account.'

Setting Up a New Repository

1. Create a directory to contain the project.
2. Go into the new directory.
3. Type `git init`.
4. Write some code.
5. Type `git add` to add the file.

git add is used to indicate that a modified file is ready to be added to the repository

6. Type `git commit`.
7. Type `git push`.

git commit adds the modifications to the repository

git push pushes committed changes to github

8. Type `git status`.

git status lists the files that have been changed and any other new additions

9. Type `git diff`.

git diff shows the exact changes made

```
anoos@DESKTOP-E89P26T MINGW64 ~
$ cd /c/Users/anoos/Documents/SampleGitRepo

anoos@DESKTOP-E89P26T MINGW64 ~/Documents/SampleGitRepo
$ git init
Initialized empty Git repository in C:/Users/anoos/Documents/SampleGitRepo/.git/

anoos@DESKTOP-E89P26T MINGW64 ~/Documents/SampleGitRepo (master)
$ ls
Hello.R

anoos@DESKTOP-E89P26T MINGW64 ~/Documents/SampleGitRepo (master)
$ git add Hello.R
```

Please supply the message using either `-m` or `-F` option.

```
anoos@DESKTOP-E89P26T MINGW64 ~/Documents/SampleGitRepo (master)
$ git commit -m "Commit Message"
[master (root-commit) acfb433] Commit Message
1 file changed, 1 insertion(+)
create mode 100644 Hello.R
```

```
anoos@DESKTOP-E89P26T MINGW64 ~/Documents/SampleGitRepo (master)
$ git remote add origin https://github.com/asagar06/SampleGitRepo.git

anoos@DESKTOP-E89P26T MINGW64 ~/Documents/SampleGitRepo (master)
$ git remote -v
origin  https://github.com/asagar06/SampleGitRepo.git (fetch)
origin  https://github.com/asagar06/SampleGitRepo.git (push)

anoos@DESKTOP-E89P26T MINGW64 ~/Documents/SampleGitRepo (master)
$ git push origin master
Counting objects: 3, done.
Writing objects: 100% (3/3), 230 bytes | 230.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/asagar06/SampleGitRepo.git
 * [new branch]      master -> master
```

Added Repository on GitHub

The screenshot shows the GitHub interface for a repository named 'SampleGitRepo' by user 'asagar06'. The repository has 1 commit, 1 branch, 0 releases, and 1 contributor. The 'Code' tab is selected, showing the commit history. The latest commit is by 'asagar06' with the message 'Commit Message' and hash 'acfb433', made 13 minutes ago. Below the commit history, there is a prompt to 'Add a README' to help people understand the project. The footer of the page shows the GitHub logo, copyright information for 2017, and links to Terms, Privacy, Security, Status, Help, Contact GitHub, API, Training, Shop, Blog, and About.

GitHub repository page for **asagar06 / SampleGitRepo**. The page shows the repository name, owner, and various statistics: 1 commit, 1 branch, 0 releases, and 1 contributor. The 'Code' tab is selected, displaying the commit history. The latest commit is by **asagar06** with the message 'Commit Message' and hash `acfb433`, made 13 minutes ago. Below the commit history, there is a prompt to 'Add a README' to help people understand the project. The footer of the page shows the GitHub logo, copyright information for 2017, and links to Terms, Privacy, Security, Status, Help, Contact GitHub, API, Training, Shop, Blog, and About.

```

anoos@DESKTOP-E89P26T MINGW64 ~/Documents/SampleGitRepo (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   Hello.R

no changes added to commit (use "git add" and/or "git commit -a")

anoos@DESKTOP-E89P26T MINGW64 ~/Documents/SampleGitRepo (master)
$ git diff
diff --git a/Hello.R b/Hello.R
index 8e23576..9f00c6a 100644
--- a/Hello.R
+++ b/Hello.R
@@ -1,2 @@
-print("Hello World")
\ No newline at end of file
+print("Hello World")
+print("Testing out Git!!!")
\ No newline at end of file

anoos@DESKTOP-E89P26T MINGW64 ~/Documents/SampleGitRepo (master)
$ git diff Hello.R
diff --git a/Hello.R b/Hello.R
index 8e23576..9f00c6a 100644
--- a/Hello.R
+++ b/Hello.R
@@ -1,2 @@
-print("Hello World")
\ No newline at end of file
+print("Hello World")
+print("Testing out Git!!!")
\ No newline at end of file

anoos@DESKTOP-E89P26T MINGW64 ~/Documents/SampleGitRepo (master)

```

Create a New Repo from an Existing Project

1. Go into the directory containing the project.
2. Type `git init`.
3. Type `git add` to add all of the relevant files.
4. If needed, create a `.gitignore` file right away, to indicate all of the files you don't want to track.
Use `git add .gitignore`, too.
5. Type `git commit`.
6. Go to GitHub.
7. Log in to your account.
8. Click the new repository button in the top-right.

```
$ git remote add origin https://github.com/username/new_repo
```

```
$ git push -u origin master
```

9. Click the "Create repository" button.

```
MINGW64:/c/Users/anoos/Documents/SampleRepo
$ ls
Greetings.txt  README.md

anoos@DESKTOP-E89P26T MINGW64 ~/Documents/Stats202A (master)
$ cd ..
\
anoos@DESKTOP-E89P26T MINGW64 ~/Documents
$ cd SampleRepo

anoos@DESKTOP-E89P26T MINGW64 ~/Documents/SampleRepo
$ git init
Initialized empty Git repository in C:/Users/anoos/Documents/SampleRepo/.git/
```

```
anoos@DESKTOP-E89P26T MINGW64 ~/Documents/SampleRepo (master)
$ git add .

anoos@DESKTOP-E89P26T MINGW64 ~/Documents/SampleRepo (master)
$ ls
Details.txt  Hello.R

anoos@DESKTOP-E89P26T MINGW64 ~/Documents/SampleRepo (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   Details.txt
        new file:   Hello.R

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        DoNotTrack.txt

anoos@DESKTOP-E89P26T MINGW64 ~/Documents/SampleRepo (master)
$ git help gitignore

anoos@DESKTOP-E89P26T MINGW64 ~/Documents/SampleRepo (master)
$ git add .gitignore
```

```

anoos@DESKTOP-E89P26T MINGW64 ~/Documents/SampleRepo (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   .gitignore
        new file:   Details.txt
        new file:   Hello.R


anoos@DESKTOP-E89P26T MINGW64 ~/Documents/SampleRepo (master)
$ git commit -m "New Repo - First Commit"
[master (root-commit) e37feaf] New Repo - First Commit
 3 files changed, 5 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 Details.txt
 create mode 100644 Hello.R

anoos@DESKTOP-E89P26T MINGW64 ~/Documents/SampleRepo (master)
$ git remote add origin https://github.com/asagar06/SampleRepo.git

anoos@DESKTOP-E89P26T MINGW64 ~/Documents/SampleRepo (master)
$ git push -u origin master
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 465 bytes | 232.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0)
To https://github.com/asagar06/SampleRepo.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.

anoos@DESKTOP-E89P26T MINGW64 ~/Documents/SampleRepo (master)
$ |

```



[Pull requests](#)
[Issues](#)
[Marketplace](#)
[Explore](#)

asagar06 / SampleRepo

Unwatch 1
Star 0
Fork 0

Code
Issues 0
Pull requests 0
Projects 0
Wiki
Insights
Settings

No description, website, or topics provided.
Edit

Add topics

1 commit
1 branch
0 releases
1 contributor


Branch: master
New pull request
Create new file
Upload files
Find file
Clone or download

asagar06
New Repo - First Commit
Latest commit e37feaf 9 minutes ago

.gitignore	New Repo - First Commit	9 minutes ago
Details.txt	New Repo - First Commit	9 minutes ago
Hello.R	New Repo - First Commit	9 minutes ago

Help people interested in this repository understand your project by adding a README.
Add a README

© 2017 GitHub, Inc.
Terms
Privacy
Security
Status
Help


[Contact GitHub](#)
[API](#)
[Training](#)
[Shop](#)
[Blog](#)
[About](#)

Contribute to Someone's Repository

1. Go to the repository on GitHub. (Say it's by myfriend, and is called the_repo, then you'll find it at http://github.com/myfriend/the_repo.)
2. Click the "Fork" button at the top right.
3. You'll now have your own copy of that repository in your GitHub account.
4. Change into that project directory (the_repo):
`$ cd the_repo`
5. Add a connection to the original owner's repository.
`$ git remote add myfriend git://github.com/myfriend/the_repo`
6. To check this remote add set up:
`$ git remote -v`
7. Make changes to files.
8. `git add` and `git commit` those changes
9. `git push` them back to GitHub. These will go to *your version* of the repository.

```
anoos@DESKTOP-E89P26T MINGW64 ~/Documents
$ git clone https://github.com/asagar06/Stats202A.git
Cloning into 'Stats202A'...
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.

anoos@DESKTOP-E89P26T MINGW64 ~/Documents
$ cd Stats202A

anoos@DESKTOP-E89P26T MINGW64 ~/Documents/Stats202A (master)
$ git remote add grrek-geek https://github.com/greek-geek/Stats202A.git

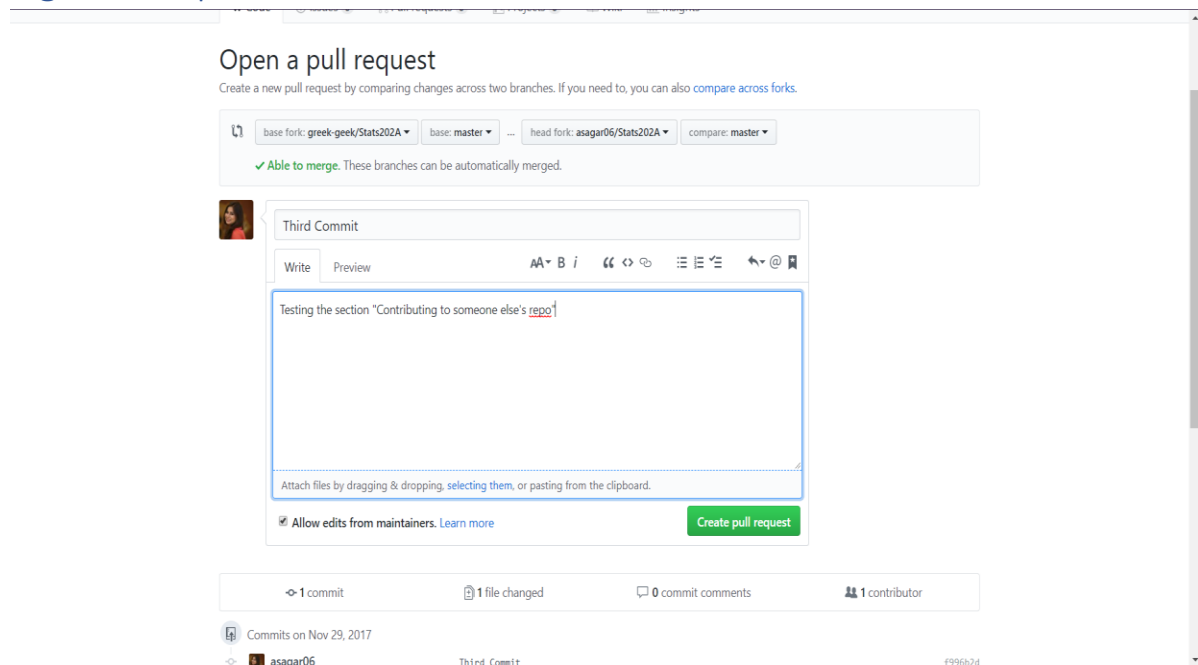
anoos@DESKTOP-E89P26T MINGW64 ~/Documents/Stats202A (master)
$ git remote -v
grrek-geek      https://github.com/greek-geek/Stats202A.git (fetch)
grrek-geek      https://github.com/greek-geek/Stats202A.git (push)
origin          https://github.com/asagar06/Stats202A.git (fetch)
origin          https://github.com/asagar06/Stats202A.git (push)

anoos@DESKTOP-E89P26T MINGW64 ~/Documents/Stats202A (master)
$ git add Greetings.txt

anoos@DESKTOP-E89P26T MINGW64 ~/Documents/Stats202A (master)
$ git commit -m "Third Commit"
[master f996b2d] Third Commit
1 file changed, 3 insertions(+)
create mode 100644 Greetings.txt

anoos@DESKTOP-E89P26T MINGW64 ~/Documents/Stats202A (master)
$ git push origin master
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 337 bytes | 337.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/asagar06/Stats202A.git
5f7991d..f996b2d master -> master
```

Creating Pull Requests



The screenshot shows the GitHub 'Open a pull request' page. At the top, it says 'Open a pull request' and 'Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).' Below this, there are dropdown menus for 'base fork: greek-geek/Stats202A', 'base: master', 'head fork: asagar06/Stats202A', and 'compare: master'. A green checkmark indicates 'Able to merge. These branches can be automatically merged.' Below this is a text area for the commit message, titled 'Third Commit'. The text area contains the text 'Testing the section "Contributing to someone else's repo"'. Below the text area is a green button labeled 'Create pull request'. At the bottom, there is a summary bar showing '1 commit', '1 file changed', '0 commit comments', and '1 contributor'. Below the summary bar, it says 'Commits on Nov 29, 2017' and 'asagar06 Third Commit f996b2d'.

Pulling Others' Changes

```
anoos@DESKTOP-E89P26T MINGW64 ~/Documents/Stats202A (master)
$ git pull grrek-geek master
From https://github.com/greek-geek/Stats202A
* branch      master      -> FETCH_HEAD
* [new branch] master      -> grrek-geek/master
Already up to date.

anoos@DESKTOP-E89P26T MINGW64 ~/Documents/Stats202A (master)
$ git push
Everything up-to-date
```

Handling Pull Requests

Using the GitHub website:

1. Go to your version of the repository.
2. Click on "Pull Requests" at the top.
3. Click on the request.

4. You'll see their comments on the pull request, and can click to see the exact changes.
5. If you want them to make further changes before you merge the changes into your repository, add a comment.
6. If you hate the whole idea, just click the "Close" button.
7. If you want to merge the changes into your repository, click the "Merge pull request" button.
8. Your GitHub repository will now be fixed, but you'll want to get them into your local repository, too.
9. Open a terminal/shell, and type

```
$ git pull
```

The screenshot shows a GitHub pull request interface for a repository named 'SampleGitRepo'. The pull request is titled 'Add files via upload #1' and is in a 'Merged' state. It shows a commit from 'bhargav265:master' merged into 'asagar06:master'. The pull request details include a conversation with comments from 'bhargav265' and 'asagar06', a commit history, and a list of files changed. The right sidebar shows metadata for the pull request, including reviewers, assignees, labels, projects, milestones, and notifications.

asagar06 / SampleGitRepo

Unwatch 1 Star 0 Fork 1

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Add files via upload #1

Merged asagar06 merged 1 commit into asagar06:master from bhargav265:master just now

Conversation 1 Commits 1 Files changed 1 +2 -0

bhargav265 commented 5 minutes ago

Added new py file

Add files via upload Verified 710eedb

asagar06 commented 2 minutes ago

Good Job Bhargav!!!

asagar06 closed this a minute ago

asagar06 reopened this just now

Reviewers: No reviews—request one

Assignees: No one—assign yourself

Labels: None yet

Projects: None yet

Milestone: No milestone

Notifications