CSc 3320: Systems Programming

Spring 2021

Homework

3: Total points 100

Submission instructions:

1. Create a Google doc for each homework assignment submission. 2. Start your responses from page 2 of the document and copy these instructions on page 1.
3. Fill in your name, campus ID and panther # in the fields provided. If this information is missing in your document TWO POINTS WILL BE DEDUCTED per submission.
4. Keep this page 1 intact on all your submissions. If this *submissions instructions* page is missing in your submission TWO POINTS WILL BE DEDUCTED per submission.
5. Each homework will typically have 2-3 PARTS, where each PART focuses on specific topic(s).
6. Start your responses to each PART on a new page.
7. If you are being asked to write code copy the code into a separate txt file and submit that as well.
8. If you are being asked to test code or run specific commands or scripts, provide the evidence of your outputs through a screenshot and copy the same into the document.
9. Upon completion, download a .PDF version of the document and submit the same.

Full Name: Kimani Guchu

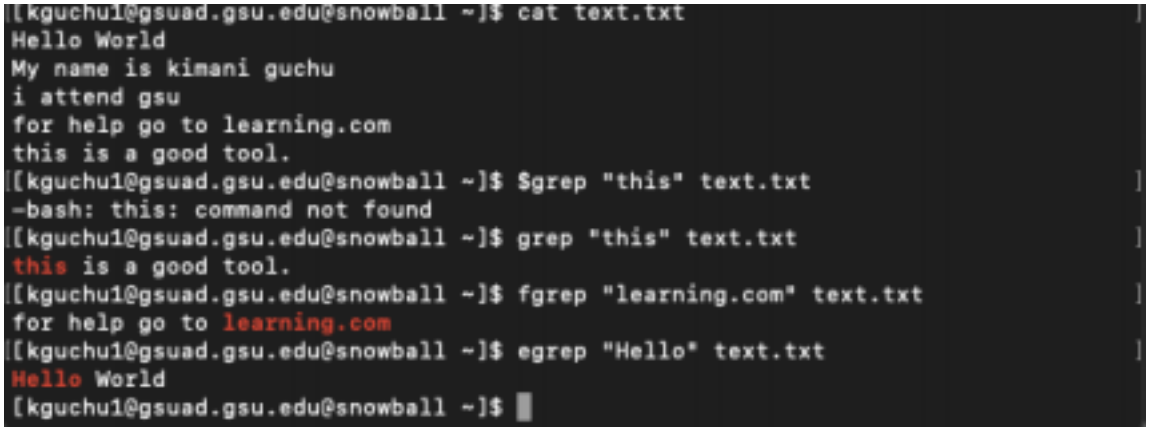
Campus ID: Kguchu1

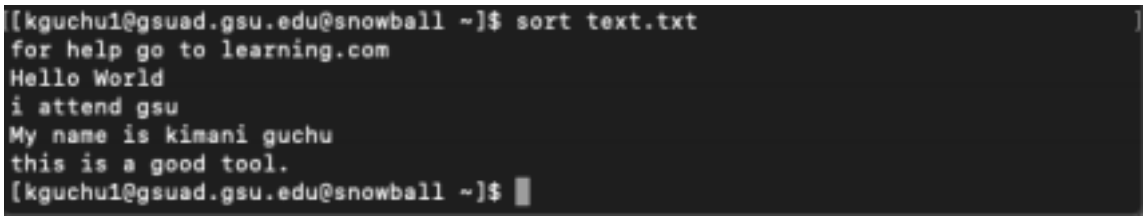
Panther #: 002-39-3714

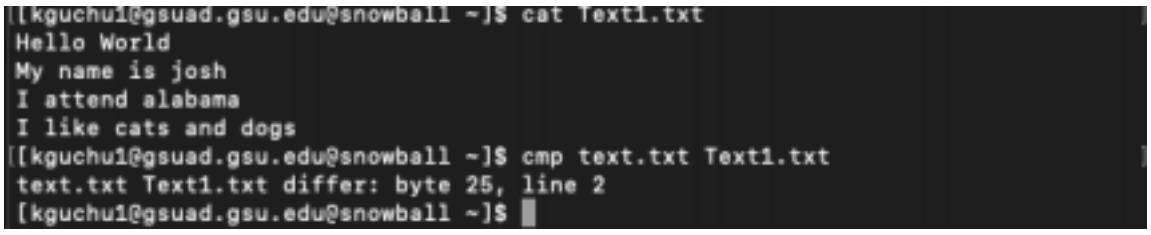
Question 1. Chapter 3

1. egrep, fgrep, grep, uniq
    - filtering files
    - grep - Allows search of regular expression
    - fgrep - allows for search fixed string only
    - Egrep - allows for a search of extended regular expression
    - Grep ex. - $grep “this” test.txt (returns all instances of “this”) - Fgrep ex -

$fgrep “learning.com” text.txt (all instances of learning.com) - Egrep ex - $egrep “Hello” text.txt (all instances of “Hello”)

- 

2. Sort
   - Sort in ascending or descending order
   - Ex. $sort test.txt (will sort file in ascending order)
   - 

3. Cmp
   - Compares if two files are the same
   - Ex. $cmp test.txt text1.txt (compares test to test1)
   - 

4. diff
   - Displays list conversion to convert one file to match another - Ex. $diff test.txt text1.txt

-
```
[kguchu1@gsuad.gsu.edu@snowball ~]$ diff text.txt Text1.txt
2,5c2,4
< My name is kimani guchu
< i attend gsu
< for help go to learning.com
< this is a good tool.
---
> My name is josh
> I attend alabama
> I like cats and dogs
[kguchu1@gsuad.gsu.edu@snowball ~]$
```

5. Find
    - Used for Locating files
    - Ex. $ find . -name ‘*.c’ -print
    - Command will find all c files in our directory
    -
```
[kguchu1@gsuad.gsu.edu@snowball ~]$ find . -name '*.c' -print
./BININT.c
./Lab3/Try.c
./Factorial.c
./myName.c
./phonee.c
./File4.c
./Int2Bin.c
./IntBin.c
./File3.c
./Test.c
./hello.c
./phonesN.c
./inbin.c
./calcPrice.c
./Binary.c
./Int2Binary.c
[kguchu1@gsuad.gsu.edu@snowball ~]$
```

6. Cpio
    - Save directories to a single backup volume
    - Ex. ls *.c | cpio -ov > backup
    - (Saves all files from our current directory into a backup)

-
```
[kguchu1@gsuad.gsu.edu@snowball ~]$ ls *.c
Binary.c     Factorial.c  hello.c       Int2Bin.c  phonee.c
BININT.c     File3.c      inbin.c       IntBin.c   phonesN.c
calcPrice.c  File4.c      Int2Binary.c  myName.c   Test.c
[kguchu1@gsuad.gsu.edu@snowball ~]$ ls *.c | cpio -ov > backup
Binary.c
BININT.c
calcPrice.c
Factorial.c
File3.c
File4.c
hello.c
inbin.c
Int2Binary.c
Int2Bin.c
IntBin.c
myName.c
phonee.c
phonesN.c
Test.c
21 blocks
[kguchu1@gsuad.gsu.edu@snowball ~]$
```

7. Tar
   - Save directories to a single backup volume -
   Ex. $tar -cvf tarfile
   - Archives current directory
   -
8. dump
   - Save directories to multiple backup volumes
9. At
   - Schedule jobs to be executed one time -
   Ex. at now + 1 minute
   - at> echo hello world > /home/kguchu1 -
   at> <EOT>
   - Outputs hello world in one minute
   -
```
[kguchu1@gsuad.gsu.edu@snowball ~]$ at now + 1 minute
at> echo hello world > /home/kguchu1
at> <EOT>
job 65 at Fri Mar  5 22:56:00 2021
[kguchu1@gsuad.gsu.edu@snowball ~]$ at -l
64      Fri Mar  5 23:13:00 2021 a kguchu1@gsuad.gsu.edu
[kguchu1@gsuad.gsu.edu@snowball ~]$
```

10. crontab

- allows you to create a scheduling table that describes a series of jobs to be executed on a periodic basis

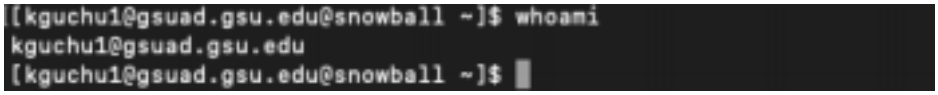
11. awk
    - scans the lines of its input and performs actions on every line that matches a particular criterion.
    - $ awk 'length($0) > 15' test.txt
    - Outputs lines over 15 characters

    -

12.Ln
    - allows you to create both hard links and symbolic (soft) links between files. - Ex. ln text.txt text
    - Creates hard link between text.txrt and a new text file

    -

    -

13.Whoami
    - Displays the name of the owner of a shell.
    - Ex $whoami
    - Retrieves the user's current ID
    - 

14. Su
    - Allows for the creation of a substitute user with privileges
    - Ex $su Kguchu1

    -

15.Biff
    - allows you to enable and disable instant mail notifications.

- Ex. $biff y (enables biff settings) then $biff (confirms)

16.compress
- replaces a file with its compressed version .z
- Ex $compress -v text.txt
- Replaces .txt with text.txt.z

17.Uncompress
- Reverses the effects of compress
- $uncompress -v text.txt.z
- Reverts back to text.txt

18.Gzip
- replaces a file with its compressed version, .gz
- Ex $gzip -v hello.c

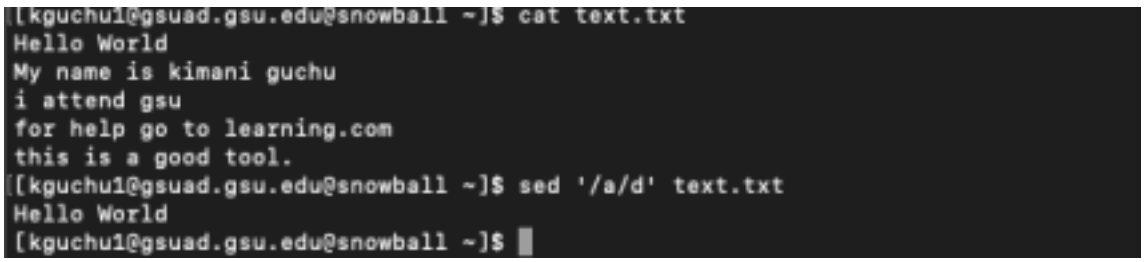
-

19.Gunzip
- gunzip can uncompress a file created by either gzip or compress. -
Ex $gunzip -v hello.c

-

20.Crypt
- creates a key-encoded version of a text file.
- Ex $crypt key < text.txt > text.crypt
- Creates an encrypted file with the key “key”
-

21.sed
- edits an input stream according to a script that contains editing commands -
Ex. $sed '/a/d' arms
- Removes all lines containing an “A”
-



22.Tr,
- maps all of the characters in its standard input from the character set
string1 to the character set string2.

- Ex. tr a-z A-Z < go.cart
- Translates all lower case letters to capital

```
[kguchu1@gsuad.gsu.edu@snowball ~]$ sed '/a/d' text.txt
Hello World
[kguchu1@gsuad.gsu.edu@snowball ~]$ tr a-z A-Z < text.txt
HELLO WORLD
MY NAME IS KIMANI GUCHU
I ATTEND GSU
FOR HELP GO TO LEARNING.COM
THIS IS A GOOD TOOL.
[kguchu1@gsuad.gsu.edu@snowball ~]$
```

-

23. Ul
    - transforms underline characters in its input so that they will be displayed correctly on the specified terminal
    - Ex $ul text.txt
    - Processes file for unlines

24.Od
    - displays the contents of fileName in a form specified by one of the following options:
    - ex . $od text.txt
    - Dumps our TEXT file octal

```
[kguchu1@gsuad.gsu.edu@snowball ~]$ od text.txt
0000000 062510 066154 020157 067527 066162 020144 046412 020171
0000020 060556 062555 064440 020163 064553 060555 064556 063440
0000040 061565 072550 064412 060440 072164 067145 020144 071547
0000060 020165 063012 071157 064040 066145 020160 067547 072040
0000100 020157 062554 071141 064556 063556 061456 066557 072012
0000120 064550 020163 071551 060440 063440 067557 020144 067564
0000140 066157 005056 057537 057537 057537 057537 057537 057537
0000160 057537 000012
0000163
```

-

25.mount
    - allows you to “splice” a device’s file system into the root hierarchy. -
    Ex. $mount
    - Displays mounted files
    -

26. umount
    - umount utility unmounts a previously mounted file system.
    - Ex $uMount (input file)
    - Will unmount user imputed file

27.Tty
    - displays the pathname of your terminal.
    - $tty

```
[kguchu1@gsuad.gsu.edu@snowball ~]$ tty
/dev/pts/5
[kguchu1@gsuad.gsu.edu@snowball ~]$
```

-

28.Time

- to report the execution time of any UNIX command
- Ex $time sort text.txt

```
[kguchu1@gsuad.gsu.edu@snowball ~]$ time sort text.txt
_____________
for help go to learning.com
Hello World
i attend gsu
My name is kimani guchu
this is a good tool.

real    0m0.003s
user    0m0.001s
sys     0m0.002s
[kguchu1@gsuad.gsu.edu@snowball ~]$
```

Question 2. Chapter 4

1.chsh

- allows you to change your default login shell -

Ex. $chsh

- User will change login

```
Changing shell for kguchu1@gsuad.gsu.edu.
New shell [/bin/bash]: chsh: Aborted.
[kguchu1@gsuad.gsu.edu@snowball ~]$
```

2. Echo

- displays its arguments to standard output -

Ex $echo hello world

- Will output hello world

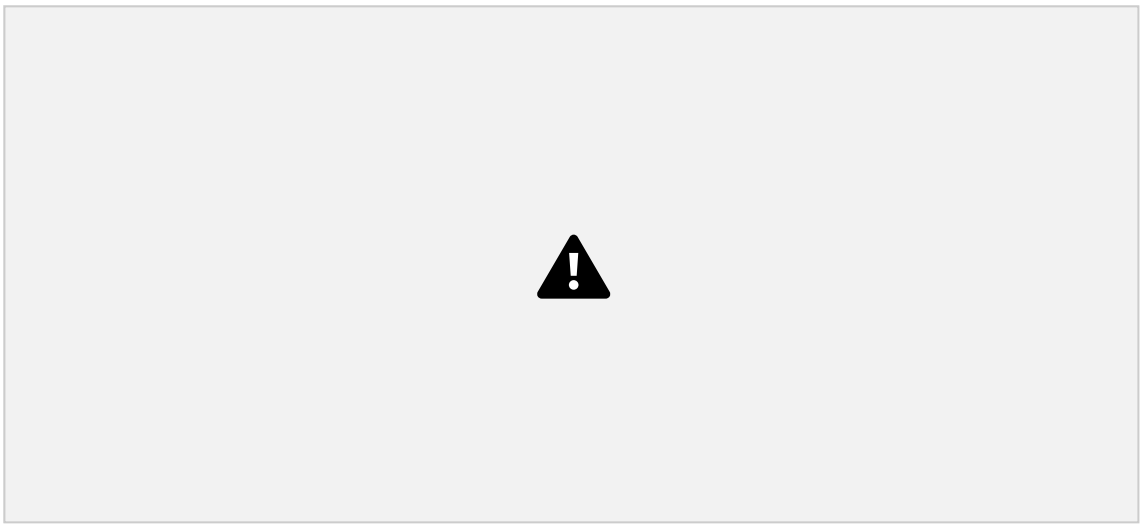
```
[kguchu1@gsuad.gsu.edu@snowball ~]$ echo hello world
hello world
[kguchu1@gsuad.gsu.edu@snowball ~]$
```

3. Kill

- to terminate a process before it is completed -

Ex $Kill -l

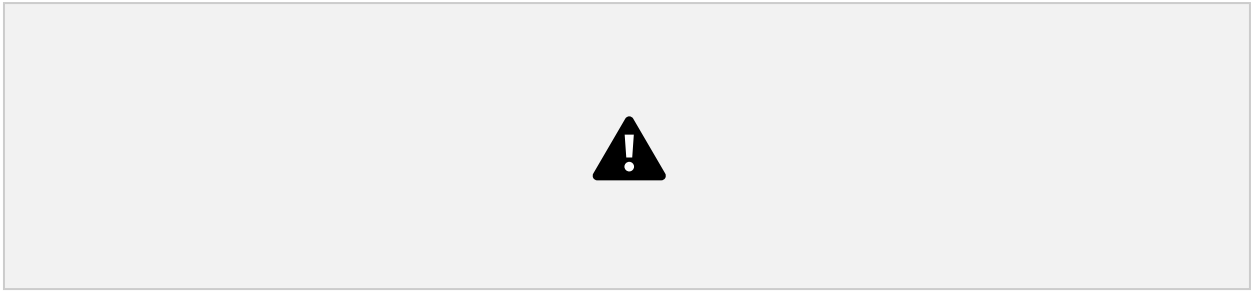
- Output running functions $kill # will kill the process -



4. Ps
   - generates a listing of process status information
   - Ex: ps -EFL

5. Nohup
   - executes the command and makes it immune to the hangup (HUP) and terminate (TERM) signals
   - Ex. $Nohup sleep 4
   - Will run a background process of sleep in the terminal
   -

6. Tee
   - copies its standard input to the specified files and to its standard output. - Ex $ who | tee who.capture | sort
   -



7. Sleep
   - sleeps for the specified number of seconds and then terminates.
   - Ex $sleep 6
   - Sleep for 6 seconds then continues
   -

8. Exec
   - causes the shell's image to be replaced with command in the process' memory space
   - Ex. $exec date
   -

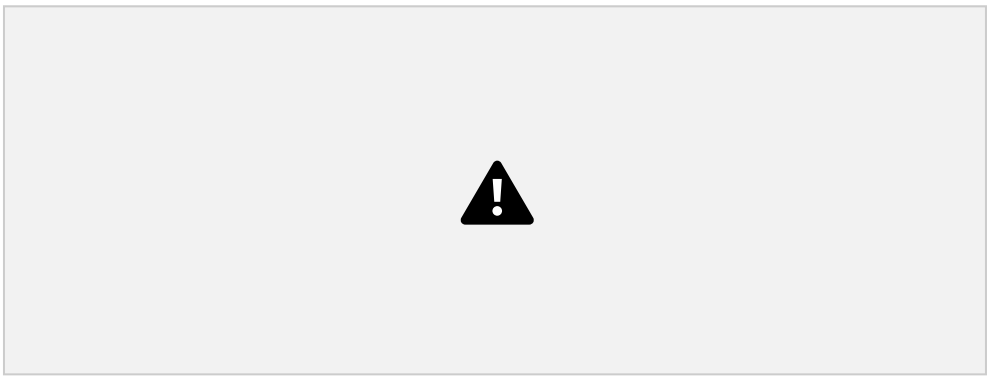
9. Unmask
   - sets the shell's umask value to the specified octal number, or displays the current umask value if the argument is omitted
   - Ex. $unmask 0
   - Sets unmask value to 0
   -

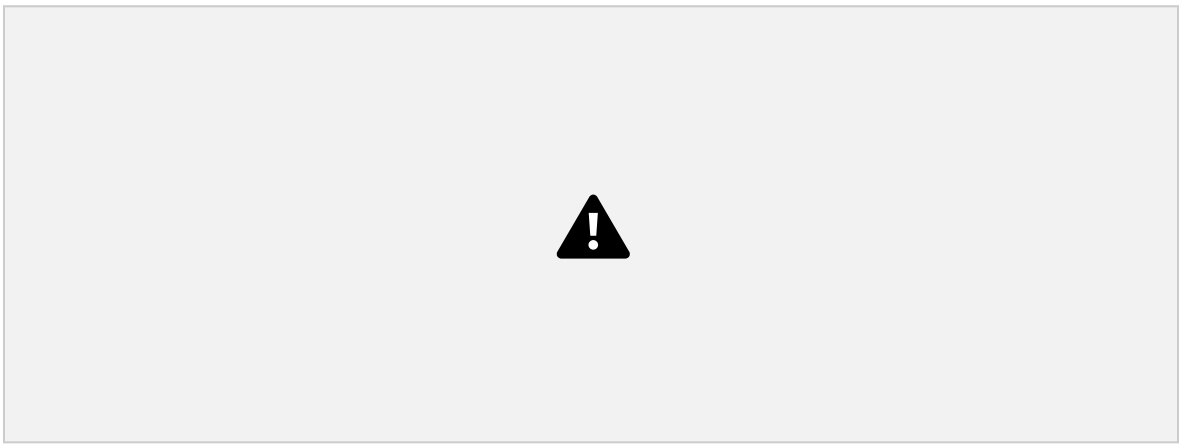
10. eval
   - executes the output of command as a regular shell command
   - Ex. $eval 'echo x=9'
   -

(Question 3 on next page)

Question 3. Chapter 5.

1. Read
    - reads one line from standard input and then assigns successive words from the line to the specified variables.
    - Ex.
    - echo "enter a number:"\n
    - read newnum
    - echo "number is $newnum"
    - Reads user imputed number
    -
2. expr
    - Supports multiple mathematical expressions
    - Ex. $expr length “hello”
    - 
3. Test
    - Determines if an expression is true or false returns 0 if false 1 if true - Ex:$ test 23 -ne 4
    - 
4. For /break
    - For if done - evaluates an expression and proceeds to given commands unless interrupted by a break statement or executes break.
    - Break - command used to exit for loop

- 

5. Case in esac
    - Case in - is an expression that evaluates to a string
    - Esac - executes if no conditions are met
    - 
6. Env
    - assigns values to specified environment variables and then executes an optional command using the new environment.
    - $ env
    - 
7. Export
    - allows you to mark local variables for export to the environment. - Ex $export
8. Readonly
    - makes the specified variables read-only
    - Ex. $readonly password

- Sets password to read-only

-

9. If / then / fi /elif
    - If - evaluates the condition
    - Then continues to the next condition until the value meets a separate condition
    - Elif - if the first evaluation is not met second evaluation is made to determine if the condition is met
    - Fi - end if, then process
    - Example script
    - x=2
    - if [$x == 10]
    - then echo "Value 10"
    - elif [$x -lt 10]
    - then
    - echo "value is less than 10"
    - fi
    -
    - 

10. Track
    - monitors the specified user's login and logout sessions.
    - Ex $track kguchu1
11. Trap
    - instructs the shell to execute command whenever any of the numbered signals signal is received
12. While / do / done
    - While- evaluates a given condition
    - Do- Expression to be executed upon the approved condition -
    Done- Exits Process, Executed if the condition is not met
    - Ex Script
    - x=3
    - while [$x -lt 4]
    - do
    - echo "hello world"

- x='expr $x+1'
- done
- exit 0
-