COEN 283 - (Operating Systems)

Programming Assignment 1

Harshkumar Pandya

Q 34)

Write a shell that is similar to Fig. 1-19 but contains enough code that it actually works so you can test it.

You might also add some features such as redirection of input and output, pipes, and background jobs.

Solution:

STEPS FOR EXECUTION:

1. CREATE A FOLDER NAMED “Folder1” ON HOME DIRECTORY AND INSIDE IT SAVE THE SHELLSCRIPT FILE AND ALSO CREATE ANOTHER FOLDER OVER THERE NAMED “Folder” WHICH CONTAINS DIFFERENT BLANK FOLDERS.
2. RUN THE SHELL SCRIPT BY TYPING FOLLOWING COMMAND ON TERMINAL:
   1. cd Folder
   2. chmod ugo+rwx myfirst1.sh
   3. ./myfirst1.sh

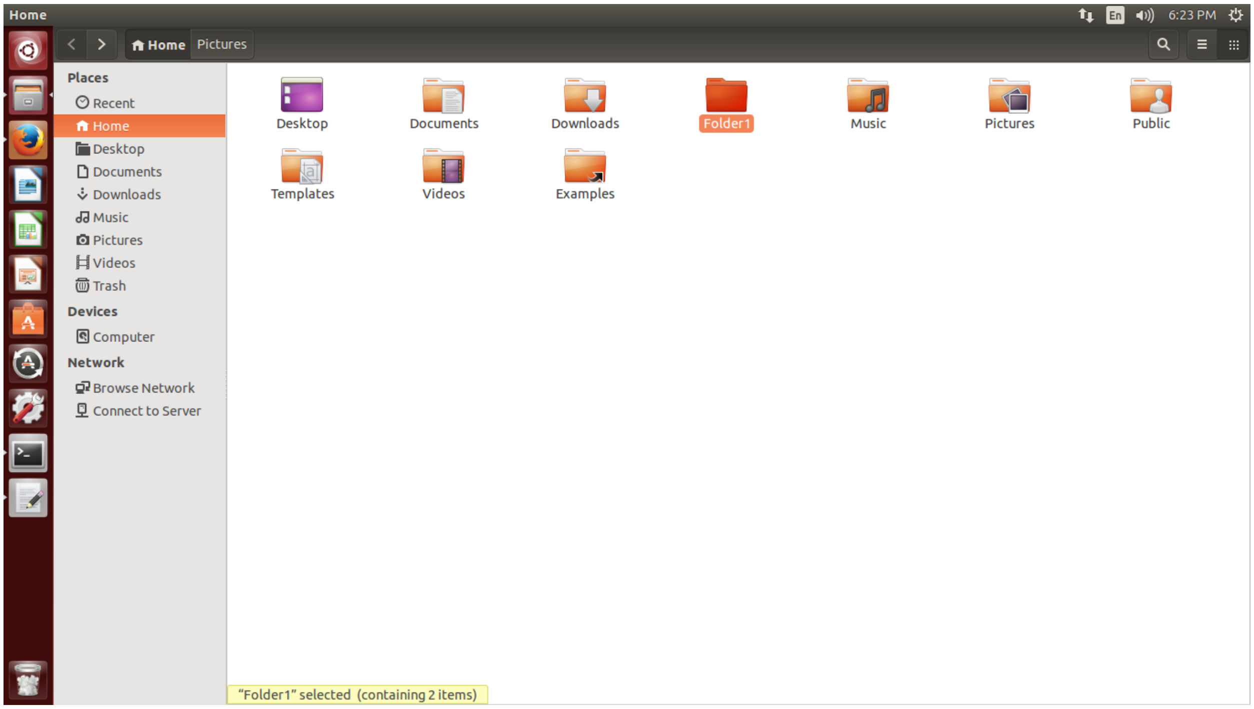
EXPLANATION:

Created various child processes. Child processes is basically a process within process. Wait function is used to wait for the child process to terminate before performing the next function. Used the ‘&’ operator so the process makes a sub process which runs in the background (daemon). Performed various functions of redirected input and output, displaying the function of "pipe" and how it is implemented using these child processes.

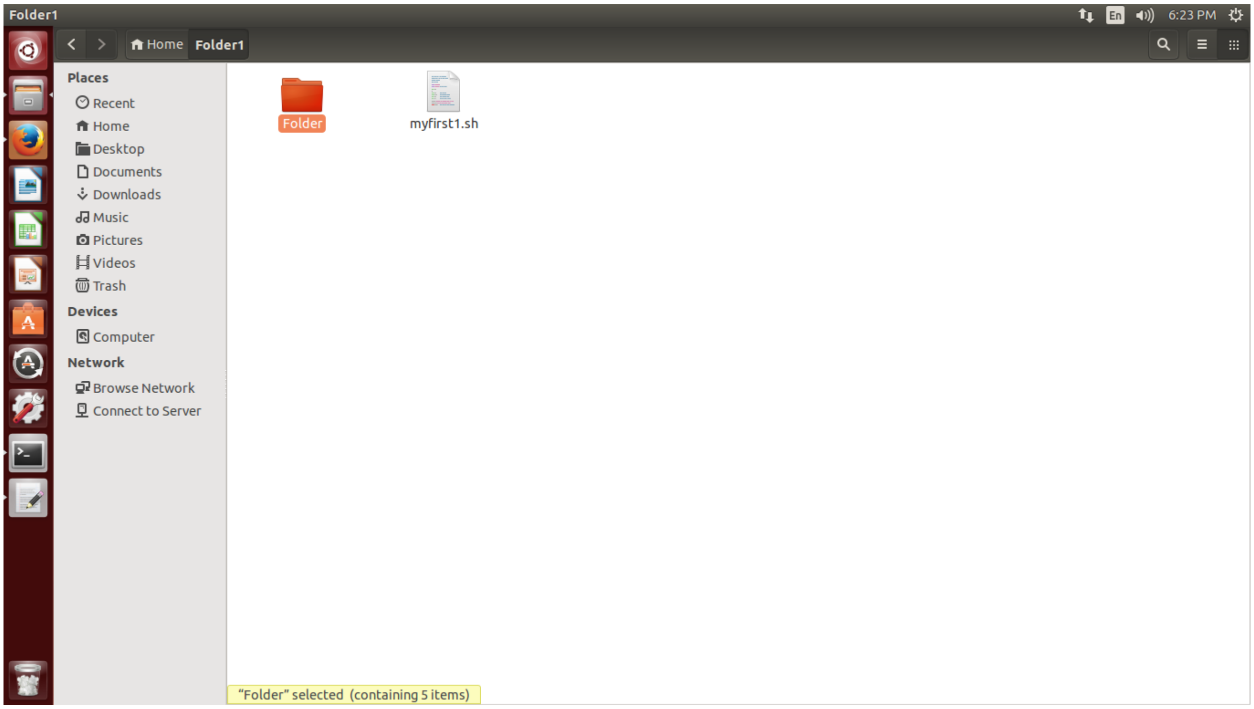
Here in my example a folder named ‘Folder1’ is created in Home which contains shell script file and another folder named ‘Folder’ which contains five empty folders (‘1ae’,’GTERTE’,’bgee2’,’je5A’ and ‘veg’)

The ‘Folder’ gets further updated when we perform the required functions.

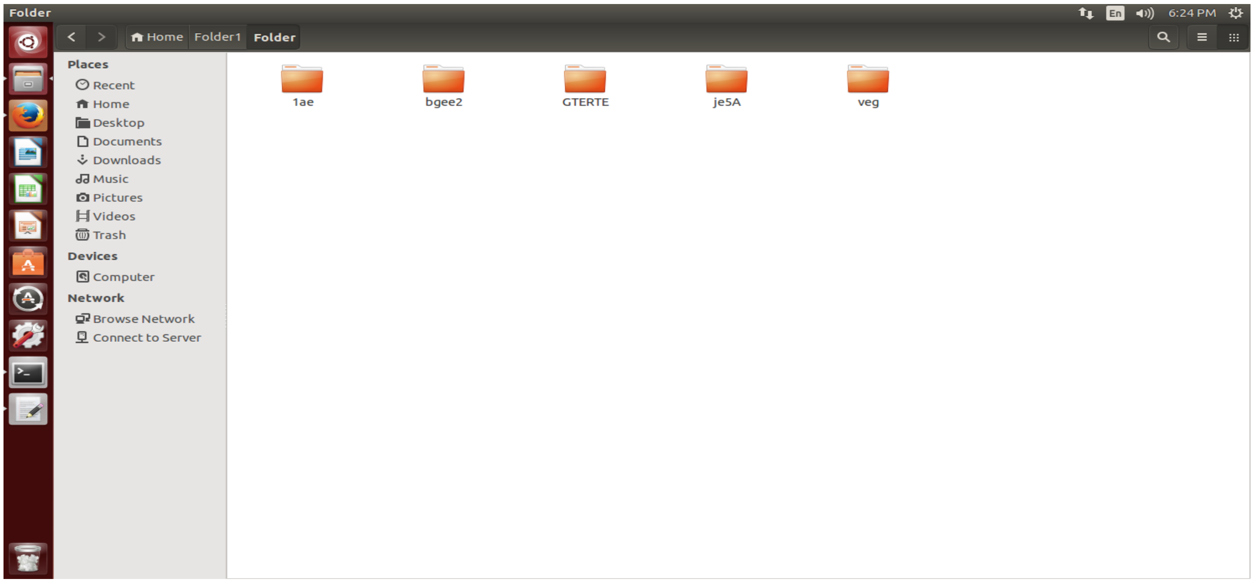
EXECUTION STEPS & OUTPUT SCREENSHOT:

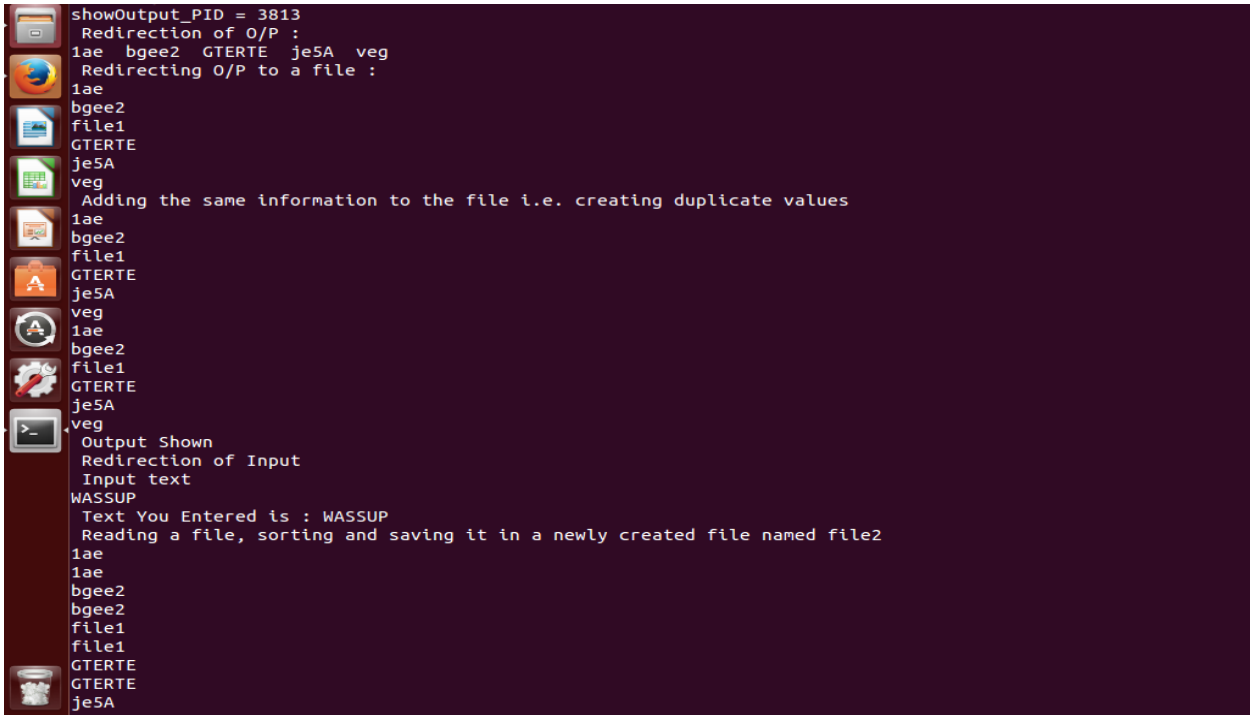


“Folder1” selected (containing 2 items)

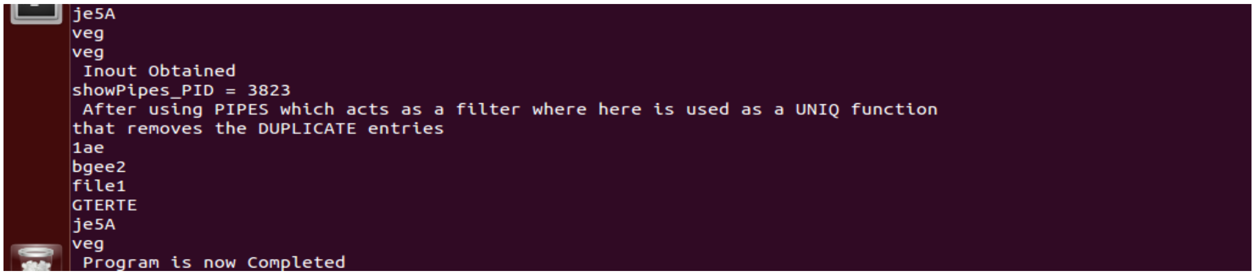


“Folder” selected (containing 5 items)

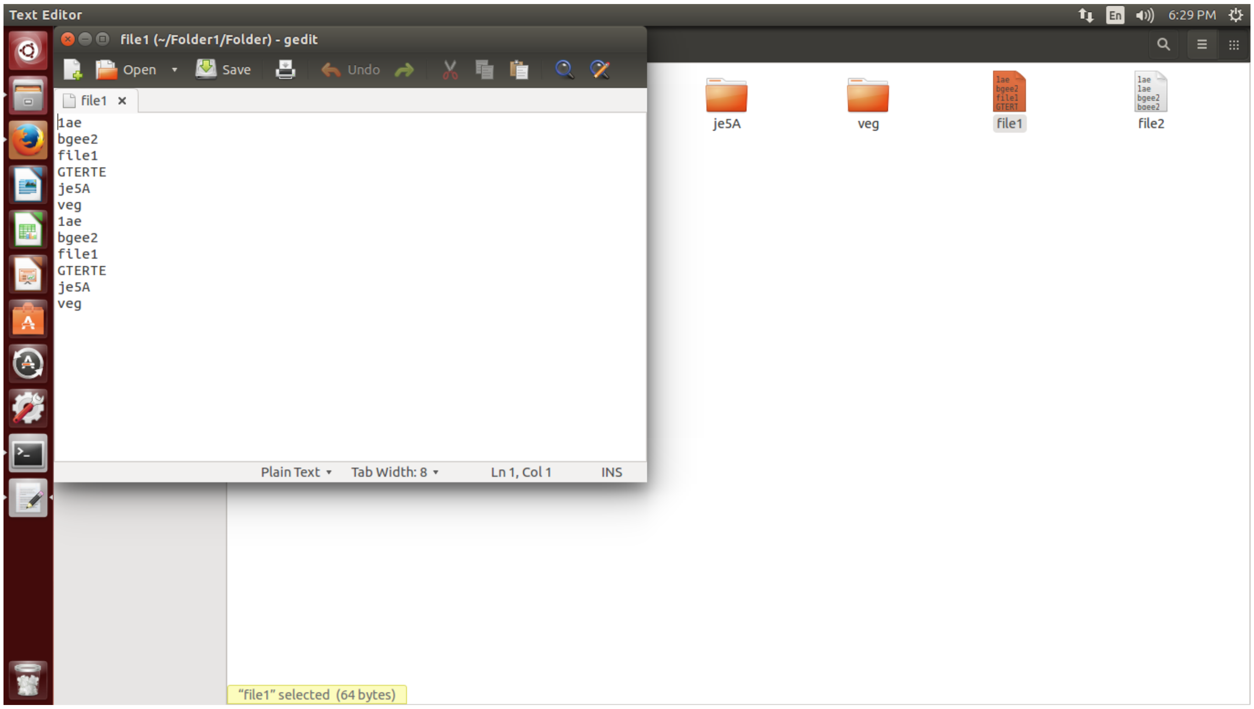




```
showOutput_PID = 3813
 Redirection of O/P :
1ae  bgee2  GTERTE  je5A  veg
 Redirecting O/P to a file :
1ae
bgee2
file1
GTERTE
je5A
veg
 Adding the same information to the file i.e. creating duplicate values
1ae
bgee2
file1
GTERTE
je5A
veg
1ae
bgee2
file1
GTERTE
je5A
veg
 Output Shown
 Redirection of Input
 Input text
WASSUP
 Text You Entered is : WASSUP
 Reading a file, sorting and saving it in a newly created file named file2
1ae
1ae
bgee2
bgee2
file1
file1
GTERTE
GTERTE
je5A
```



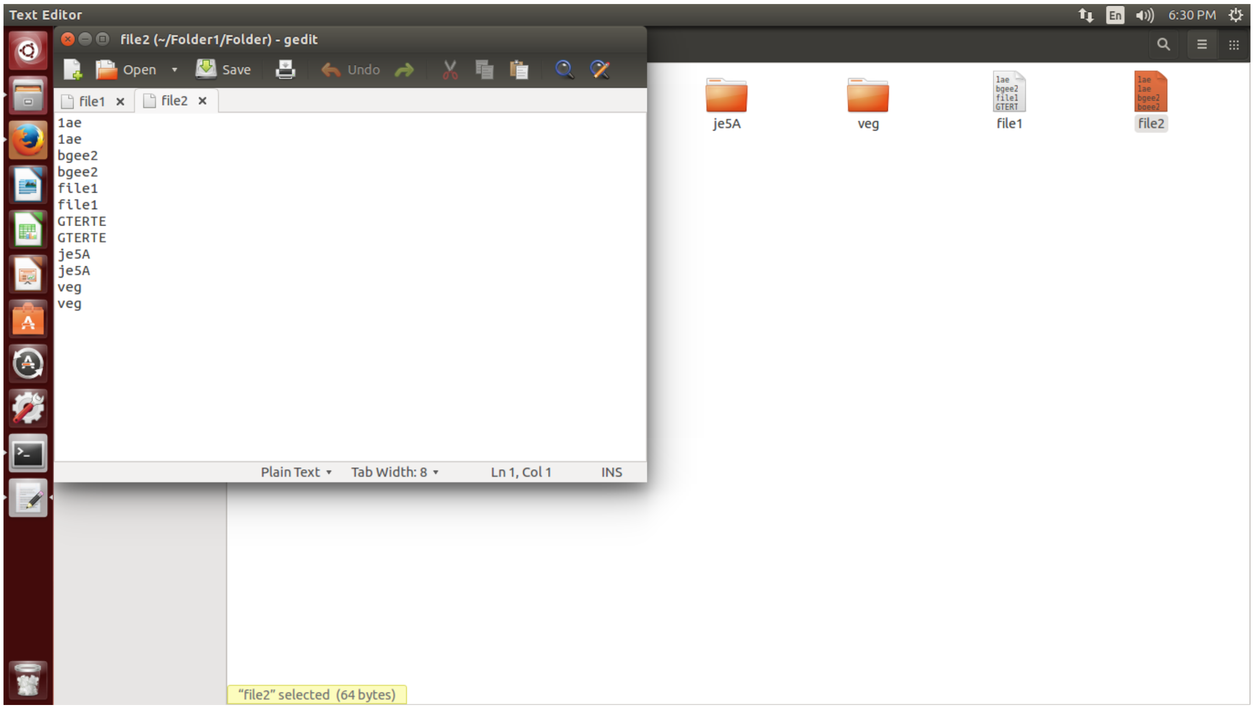
```
je5A
veg
veg
 Inout Obtained
showPipes_PID = 3823
 After using PIPES which acts as a filter where here is used as a UNIQ function
that removes the DUPLICATE entries
1ae
bgee2
file1
GTERTE
je5A
veg
 Program is now Completed
```



```
1ae
bgee2
file1
GTERTE
je5A
veg
1ae
bgee2
file1
GTERTE
je5A
veg
```

Plain Text   Tab Width: 8   Ln 1, Col 1   INS

“file1” selected (64 bytes)



```
1ae
1ae
bgee2
bgee2
file1
file1
GTERTE
GTERTE
je5A
je5A
veg
veg
```

Plain Text   Tab Width: 8   Ln 1, Col 1   INS

“file2” selected (64 bytes)

PROGRAM:

```
#!/bin/bash
DisplayOP()
{
cd Folder
echo " Redirection of O/P :"

ls                                  # Contents inside Folder1
echo " Redirecting O/P to a file :"
ls > file1                          # O/P is saved in the file named List
cat file1                           # Display contents inside List file
echo " Adding the same information to the file i.e. creating duplicate values "
ls >> file1                         # Append duplicate entries
cat file1
}
RecieveINP()
{
cd Folder
echo " Redirection of Input"
echo " Input text "
read text                           # Reading from Terminal
echo " Text You Entered is : $text "
echo " Reading a file, sorting and saving it in a newly created file named file2 "
sort < file1 > file2                # Read from File sort and then save it to file2
cat file2
}
showPipes()
```

```
{
# Pipes are also used as Filters

echo " After using PIPES which acts as a filter where here is used as a UNIQ function that removes the DUPLICATE entries"
cat file2 | uniq

# '|' pipes the output of Newlist to 'uniq'- which deletes all the repeated entries.
}



#Execution starts
DisplayOP&
echo showOutput_PID = $!                        # Prints the Process ID of the executed Child Process
wait
# Suspends execution of the calling process till the created child thread or the O/P method terminates.
echo " Output Shown "                           # Displayed after the child process completion
RecieveINP                                      # get call to the Input
echo " Inout Obtained "
showPipes&                                      # show use of pipes
echo showPipes_PID = $!
wait
echo " Program is now Completed "
```