```bash
AvikRnB:~ avikrb$ cat processwatcher
#!/bin/bash

## AUTHOR: AVIK RANJAN BHATTARAI
## DESCRIPTION: SCRIPT THAT WILL FILTER OUTPUT OF PS COMMAND ACCORDING TO OPTION
## COURSE: SCRIPT PROGRAMMING FOR NETWORK ADMINISTRATION
## INSTITUTE: UNIVERSITY OF NEW HAVEN
## DATE: 02/11/2014

echo
echo


##FUNCTION TO DISPLAY PROCESSES INDICATED BY A CERTAIN USER
function user
{
        echo "What is your user ID: "
        read id
        ps -e -o user -e -o pid -e -o pcpu -e -o pmem -e -o rss -e -o vsz > temptext
        ##COMMAND STORES THE USER, PID, %CPU, %MEM AND RSS COLUMNS INTO A FILE CALLED TEMPTEXT
        ##YOU WILL HAVE TO CREATE FILE TEMPTEXT BEFORE RUNNING THE SCRIPT
        grep -i "user" temptext
        grep -i "$id" temptext
        ##COMMAND PRINTS THE PROCESSES OF ONLY THE ID STORED IN $id

}


##FUNCTION TO DISPLAY ONLY THOSE PROCESSES WITH CERTAIN CPU USAGE
function percentage
{
        ps -e -o user -e -o pid -e -o pcpu > temptext
        ##COMMAND STORES THE USER, PID, AND %CPUCOLUMNS INTO A FILE CALLED TEMPTEXT
        grep -v -e "0.0" temptext
        ##COMMAND PRINTS THOSE PROCESSES NOT HAVING 0.0% CPU USAGE WHICH WERE STORED IN TEMPTEXT

}


##FUNCTION TO DISPLAY THOSE PROCESSES WITH CERTAIN MEMORY USAGE
function memory
{
        ps -e -o user -e -o pid -e -o pmem -e -o rss > temptext
        ##COMMAND STORES USER, PID, %MEM AND RSS (IN KILOBYTES) INTO A FILE CALLED TEMPTEXT
        grep -v -e "0.0" temptext
        ##COMMAND PRINTS THOSE PROCESSES NOT HAVING 0.0% MEMORY USAGE WHICH WERE STORED IN TEMPTEXT

}


##FUNCTION TO DISPLAY WHETHER A USER TYPED PROCESSES IS A BAD PROCESS OR NOT
##FUNCTION ALSO GIVES AN OPTION TO KILL IF THE PROCESS IS A BAD PROCESS
function badprocess
{
echo "Enter Process name to check if it is a bad Process: "
read badfile
if [ $badfile == `grep -i "$badfile" badprocess.txt` ]
##THE IF STATEMENT CHECKS THE ENTRY IN $BADFILE AND TRYS TO MATCH WITH ANY OF THE FILENAMES LISTED IN BADPROCESS.TXT
##BADPROCESS.TXT IS A FILE THAT CONTAINS LIST OF RANDOMLY GENERATED FILENAMES BY AVIK RANJAN BHATTARAI
then
    echo " $badfile is a bad process"
    echo "Do you want to stop $badfile [Y/N]??"
    read answer
        if [ $answer  == "y" ] || [ $answer == "Y" ]
            then
            killall -9 $badfile
                        ##KILLALL -9 COMMAND TERMINATES PROCESS IN A NON CACHEABLE AND NON IGNORABLE WAY
            elif [ $answer == "n" ] || [ $answer == "N" ]
                        ##IF THE USER CHOOSES TO NOT KILL THE BAD PROCESS THEN THE PROCESS IS STILL RUNNING
```

```
                then
                echo "$badfile Process is still running!!!!"
                else
                echo
                echo
                echo "****Bad entry*****"
            fi
        else
        echo "$badfile is not a bad process"
            ##IF THE PROCESS NAME IS NOT CONTAINED IN FILE BADPROCESS.TXT THEN THAT PROCESS IS NOT A BAD PROCESS
fi
}




##MAIN SCRIPT PROGRAM, DISPLAYS MENU OPTIONS TO CHOOSE

echo "Enter your Choice of displaying Processes:  "
echo "1.   User account"
echo "2.   Percentage of CPU"
echo "3.   Memory"
echo "4.   Look for bad processes"
read choice
if [ $choice == 1 ];
        then
        user
        ##CALL FUNCTION USER

        elif [ $choice == 2 ];
        then
                percentage
                ##CALL FUNCTION PERCENTAGE

                        elif [ $choice == 3 ];
                        then
                                memory
                                ##CALL FUNCTION MEMORY

                                elif [ $choice == 4 ];
                                then
                                        badprocess
                                        ##CALL FUNCTION BADPROCESS
fi

AvikRnB:~ avikrb$
```