BASH COURSEWORK

-MAHEK TRIVEDI M00979199

Contents

Need For Project

- Helps to create an understanding of the fundamentals of Linux bash scripting.
- Helps beginners in bash scripting enhance their basic knowledge and skills.
- Develops research and trial and error(experimentation) skills

Functions Part 1

show_calender
shows calender and saves the selected date in a variable and for the selected date , we can enter a note

show_date_time : Displays the current date and time using dialog's infobox format.

FUNCTIONS (Part 1)

delete_file() -Ask's user to enter a directory name, list all files present in the directory, and delete the file chosen by the user.

Functions Part 2

show_os_info()
Displays OS(Linux Standard Base) information

show_cpu_info()
Displays information about CPU Architecture

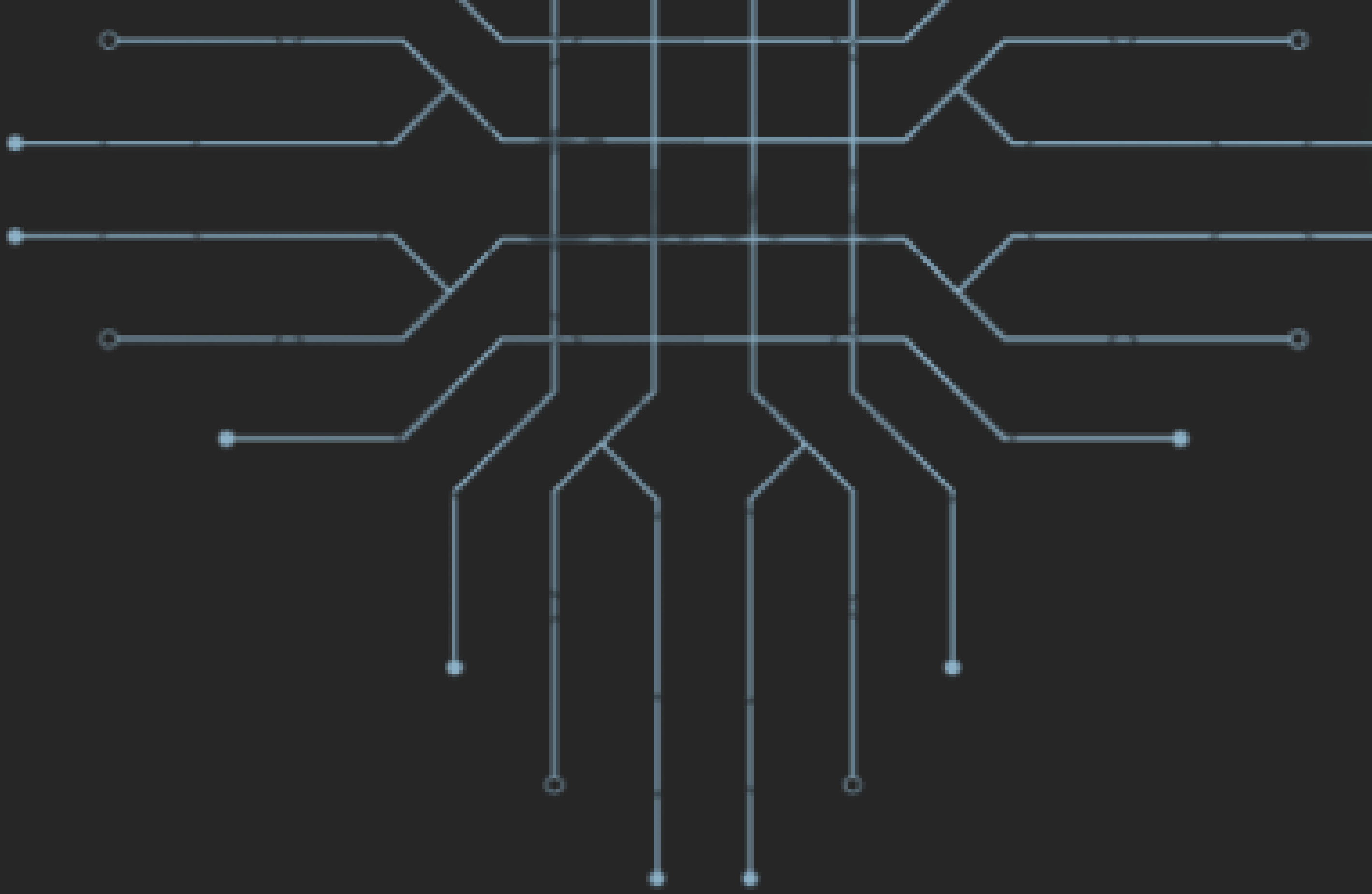
FUNCTIONS (Part 2)

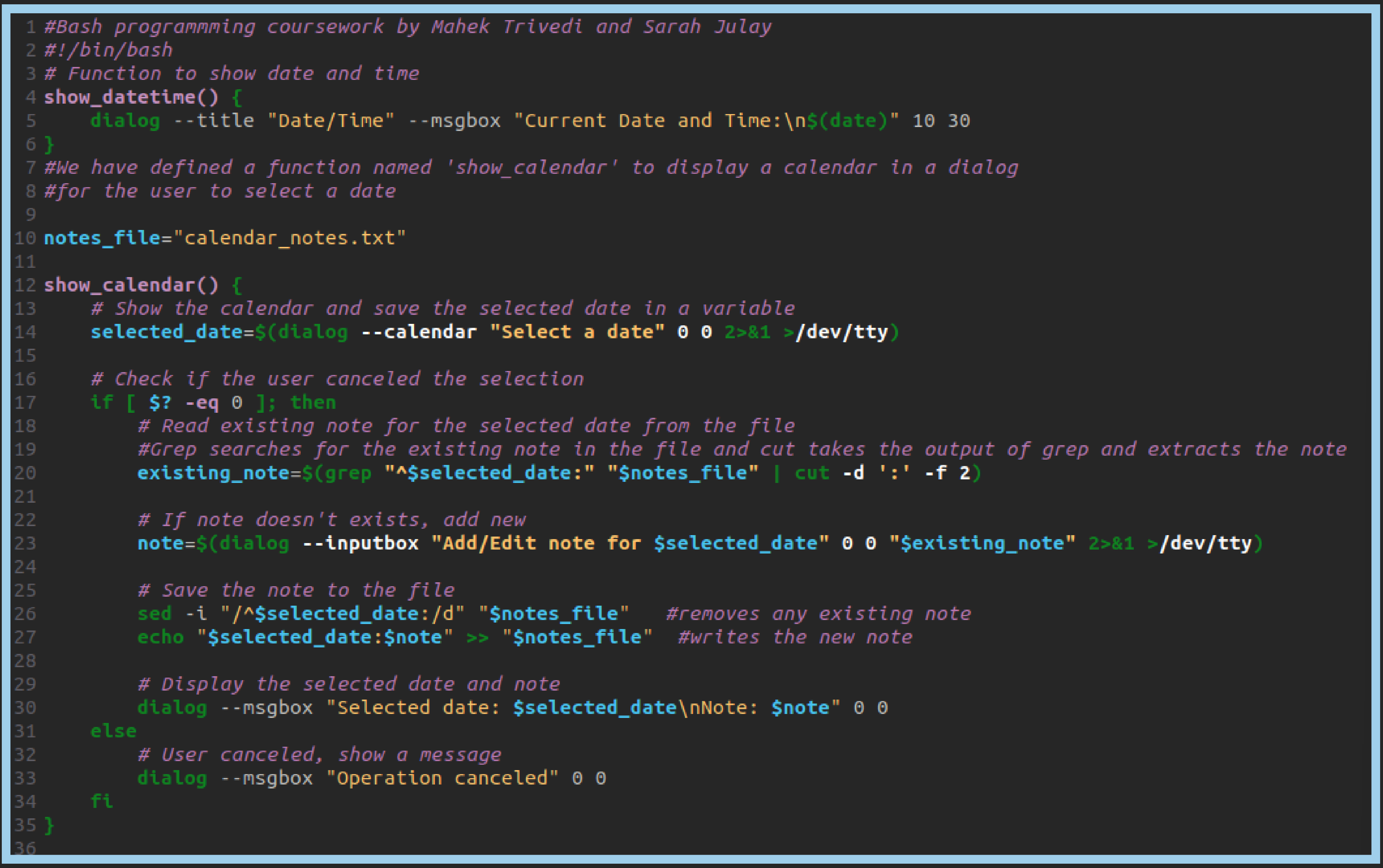
show_disk_info()
Displays information about the disk and it's partitions

show_file_system_info()
Displays information about the files and where they are mounted

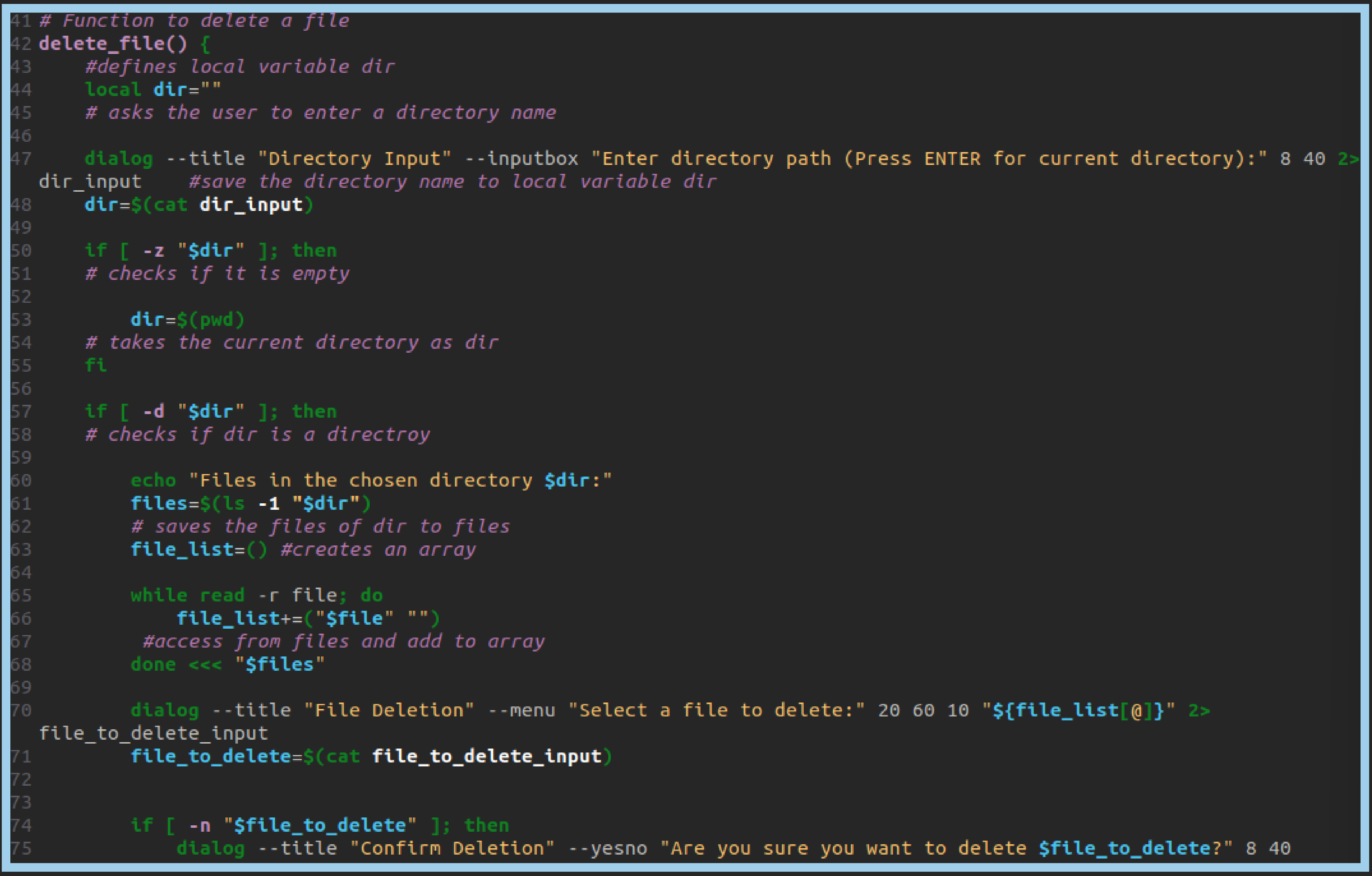
show_memory_info()
Displays information about the memory. The output is in gigabites

Code Part 1





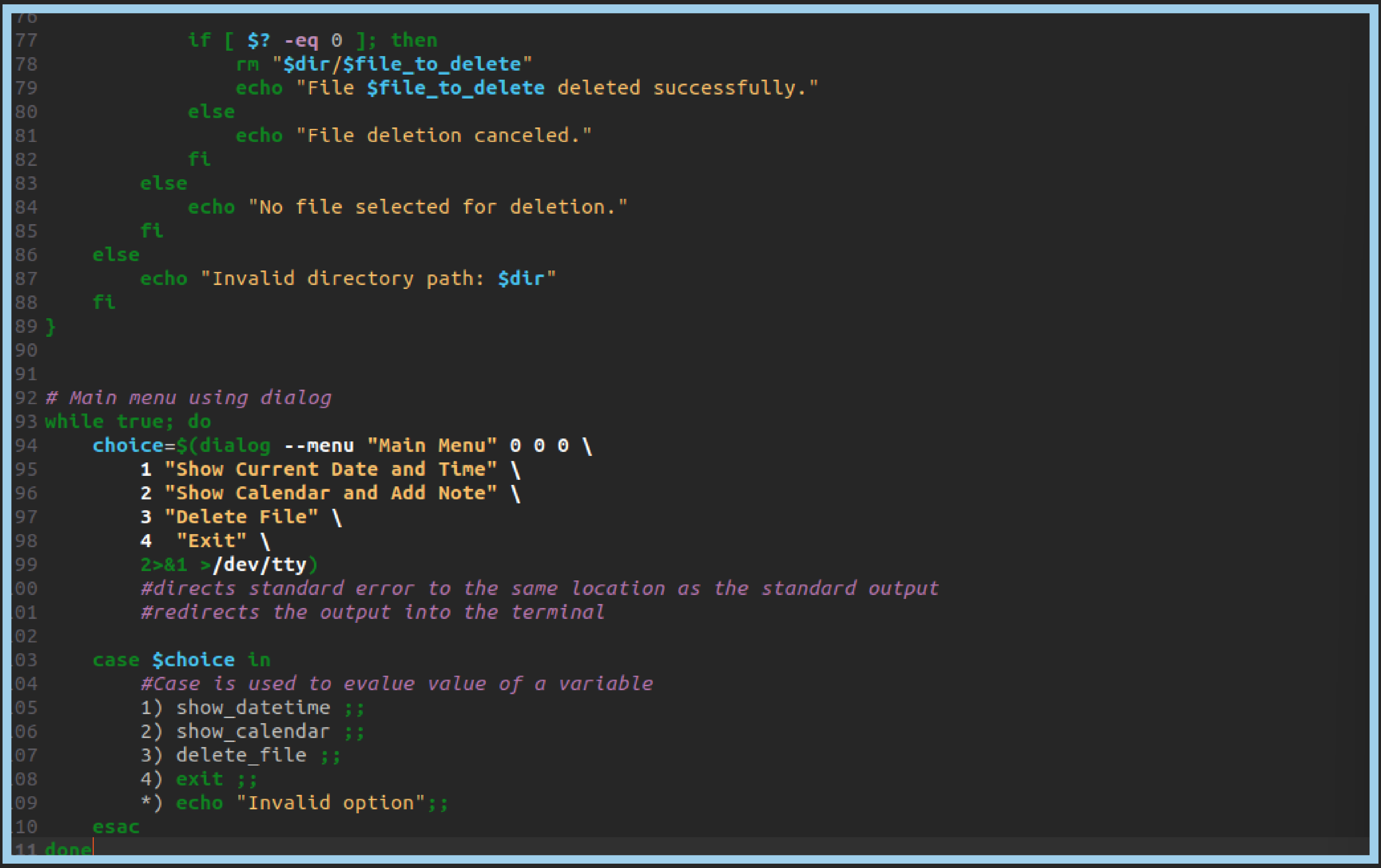
```bash
#Bash programmming coursework by Mahek Trivedi and Sarah Julay
#!/bin/bash
# Function to show date and time
show_datetime() {
    dialog --title "Date/Time" --msgbox "Current Date and Time:\n$(date)" 10 30
}
#We have defined a function named 'show_calendar' to display a calendar in a dialog
#for the user to select a date

notes_file="calendar_notes.txt"

show_calendar() {
    # Show the calendar and save the selected date in a variable
    selected_date=$(dialog --calendar "Select a date" 0 0 2>&1 >/dev/tty)

    # Check if the user canceled the selection
    if [ $? -eq 0 ]; then
        # Read existing note for the selected date from the file
        #Grep searches for the existing note in the file and cut takes the output of grep and extracts the note
        existing_note=$(grep "^$selected_date:" "$notes_file" | cut -d ':' -f 2)

        # If note doesn't exists, add new
        note=$(dialog --inputbox "Add/Edit note for $selected_date" 0 0 "$existing_note" 2>&1 >/dev/tty)

        # Save the note to the file
        sed -i "/^$selected_date:/d" "$notes_file"   #removes any existing note
        echo "$selected_date:$note" >> "$notes_file"  #writes the new note

        # Display the selected date and note
        dialog --msgbox "Selected date: $selected_date\nNote: $note" 0 0
    else
        # User canceled, show a message
        dialog --msgbox "Operation canceled" 0 0
    fi
}

```



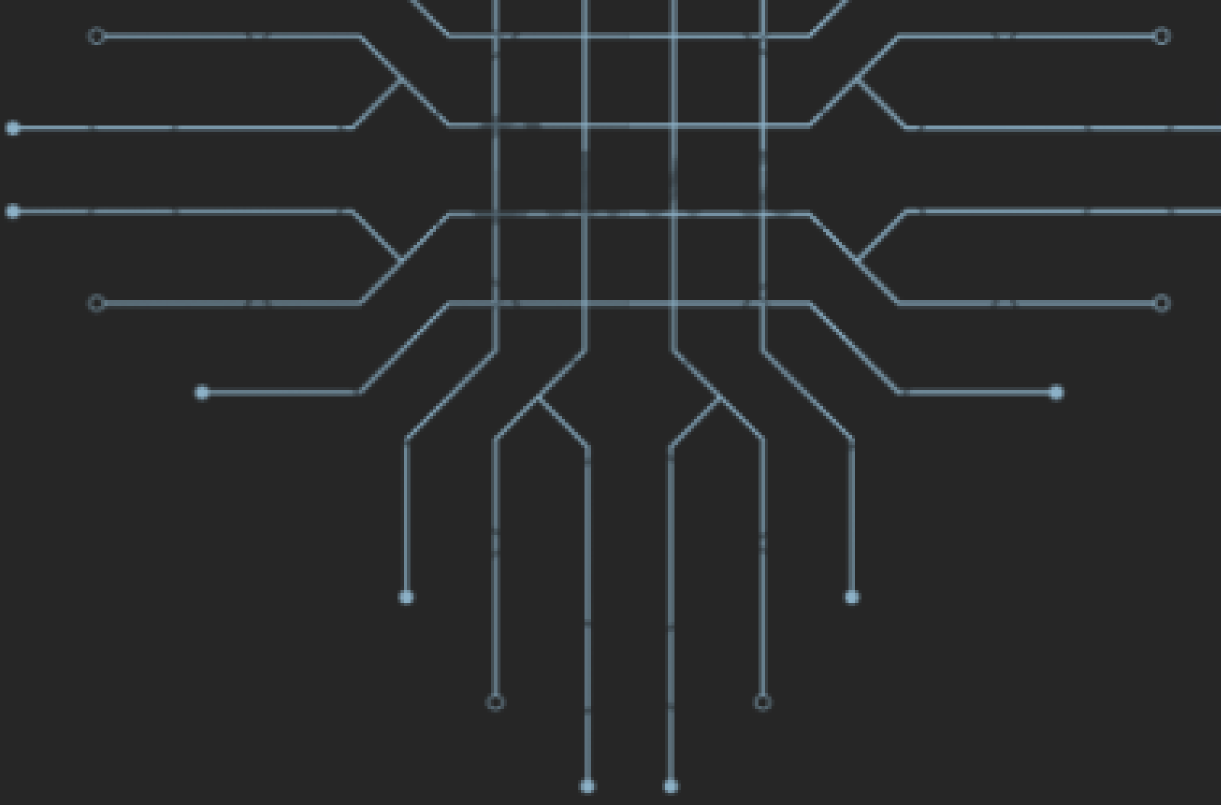
```bash
# Function to delete a file
delete_file() {
    #defines local variable dir
    local dir=""
    # asks the user to enter a directory name

    dialog --title "Directory Input" --inputbox "Enter directory path (Press ENTER for current directory):" 8 40 2>
dir_input    #save the directory name to local variable dir
    dir=$(cat dir_input)

    if [ -z "$dir" ]; then
    # checks if it is empty

        dir=$(pwd)
    # takes the current directory as dir
    fi

    if [ -d "$dir" ]; then
    # checks if dir is a directroy

        echo "Files in the chosen directory $dir:"
        files=$(ls -1 "$dir")
        # saves the files of dir to files
        file_list=() #creates an array

        while read -r file; do
            file_list+=("$file" "")
         #access from files and add to array
        done <<< "$files"

        dialog --title "File Deletion" --menu "Select a file to delete:" 20 60 10 "${file_list[@]}" 2>
file_to_delete_input
        file_to_delete=$(cat file_to_delete_input)


        if [ -n "$file_to_delete" ]; then
            dialog --title "Confirm Deletion" --yesno "Are you sure you want to delete $file_to_delete?" 8 40
```

Code Part 1

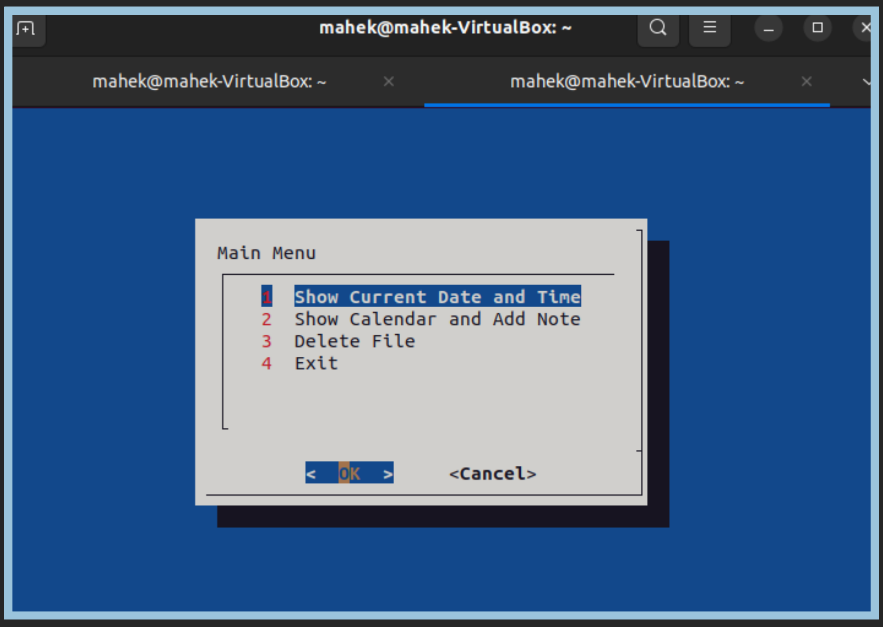
continued...



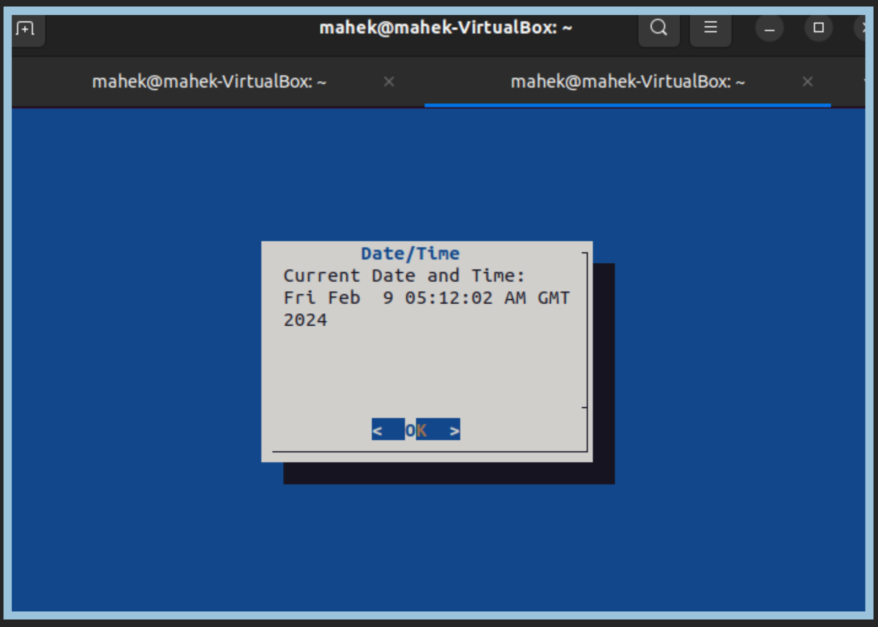
```
            if [ $? -eq 0 ]; then
                rm "$dir/$file_to_delete"
                echo "File $file_to_delete deleted successfully."
            else
                echo "File deletion canceled."
            fi
        else
            echo "No file selected for deletion."
        fi
    else
        echo "Invalid directory path: $dir"
    fi
}


# Main menu using dialog
while true; do
    choice=$(dialog --menu "Main Menu" 0 0 0 \
        1 "Show Current Date and Time" \
        2 "Show Calendar and Add Note" \
        3 "Delete File" \
        4  "Exit" \
        2>&1 >/dev/tty)
        #directs standard error to the same location as the standard output
        #redirects the output into the terminal

    case $choice in
        #Case is used to evalue value of a variable
        1) show_datetime ;;
        2) show_calendar ;;
        3) delete_file ;;
        4) exit ;;
        *) echo "Invalid option";;
    esac
done
```



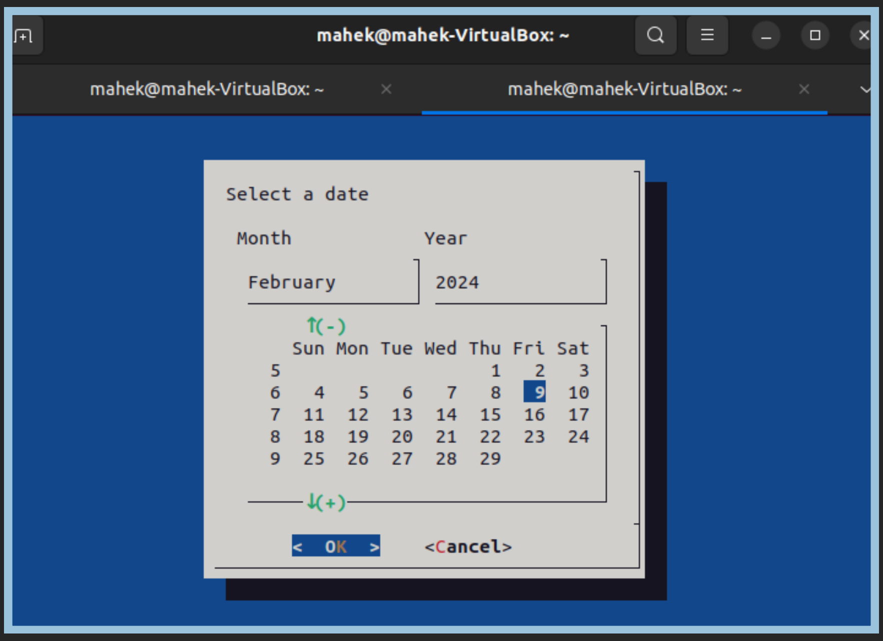
Output part 1


mahek@mahek-VirtualBox: ~
mahek@mahek-VirtualBox: ~
mahek@mahek-VirtualBox: ~

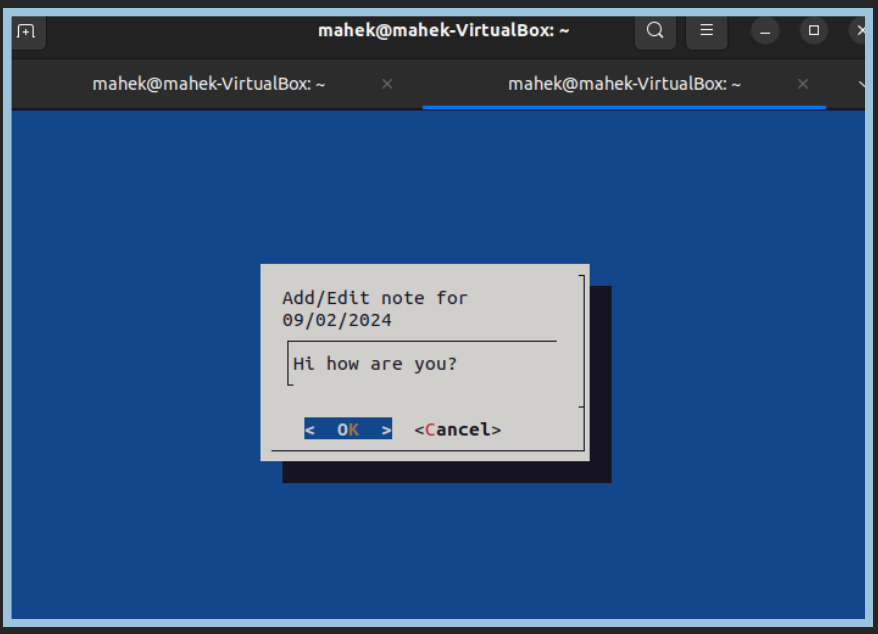
```
Main Menu

 1  Show Current Date and Time
 2  Show Calendar and Add Note
 3  Delete File
 4  Exit

        <  OK  >        <Cancel>
```


mahek@mahek-VirtualBox: ~
mahek@mahek-VirtualBox: ~
mahek@mahek-VirtualBox: ~

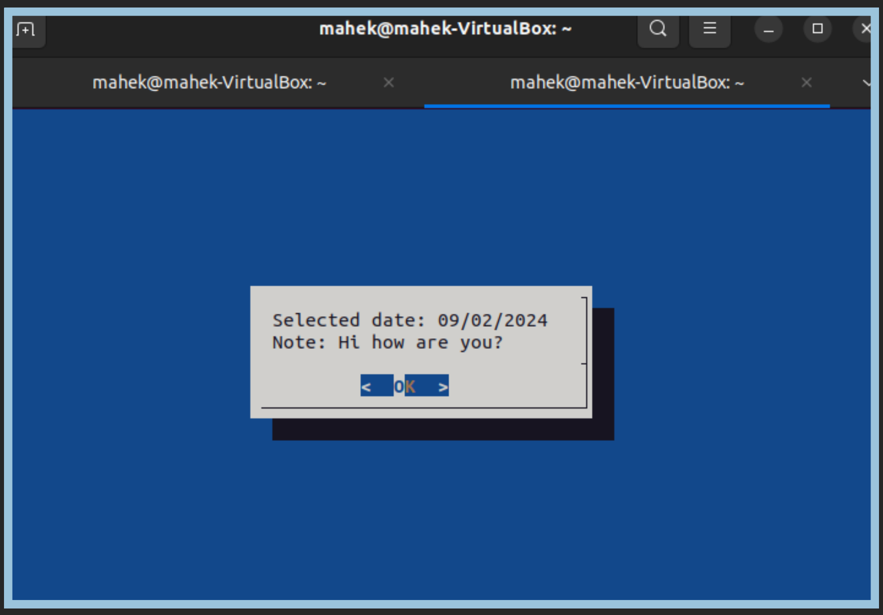
```
        Date/Time
Current Date and Time:
Fri Feb  9 05:12:02 AM GMT
2024

         < OK >
```


mahek@mahek-VirtualBox: ~
mahek@mahek-VirtualBox: ~
mahek@mahek-VirtualBox: ~

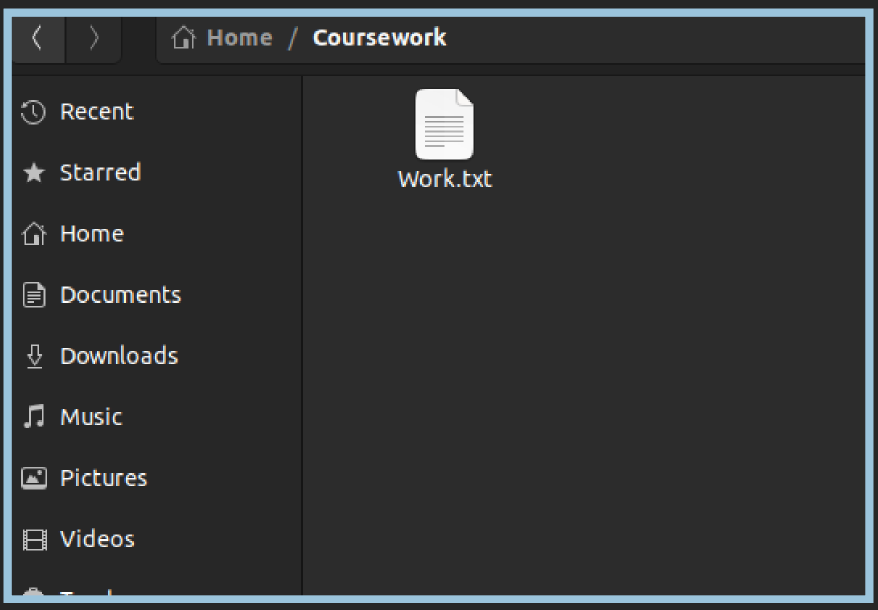
```
Select a date

 Month            Year

  February         2024

      ↑(-)
    Sun Mon Tue Wed Thu Fri Sat
 5                       1   2   3
 6   4   5   6   7   8   9  10
 7  11  12  13  14  15  16  17
 8  18  19  20  21  22  23  24
 9  25  26  27  28  29
      ↓(+)

      <  OK  >     <Cancel>
```


mahek@mahek-VirtualBox: ~
mahek@mahek-VirtualBox: ~
mahek@mahek-VirtualBox: ~

```
Add/Edit note for
09/02/2024

 Hi how are you?

  <  OK  >   <Cancel>
```

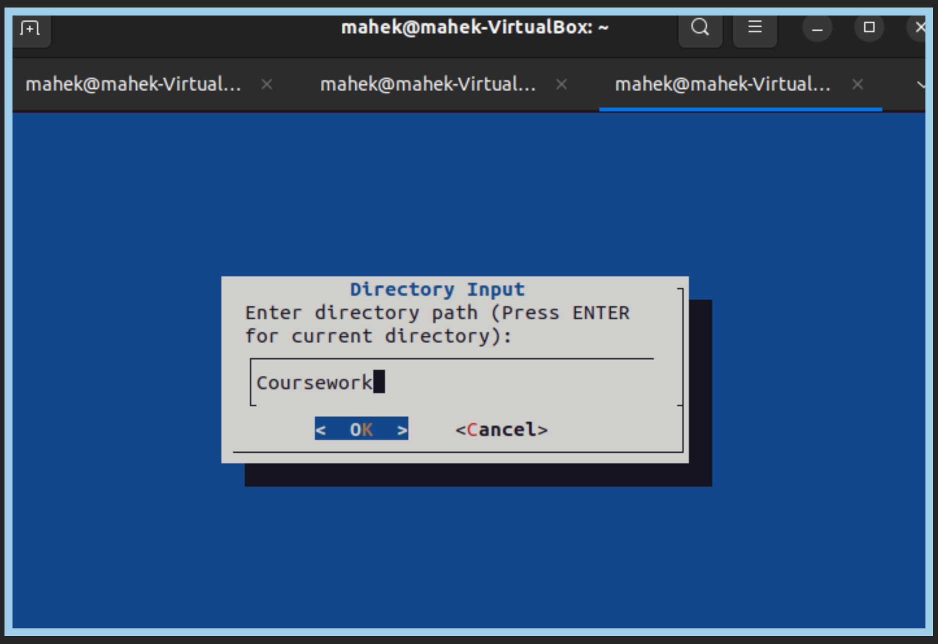

mahek@mahek-VirtualBox: ~
mahek@mahek-VirtualBox: ~
mahek@mahek-VirtualBox: ~

```
Selected date: 09/02/2024
Note: Hi how are you?

        < OK >
```

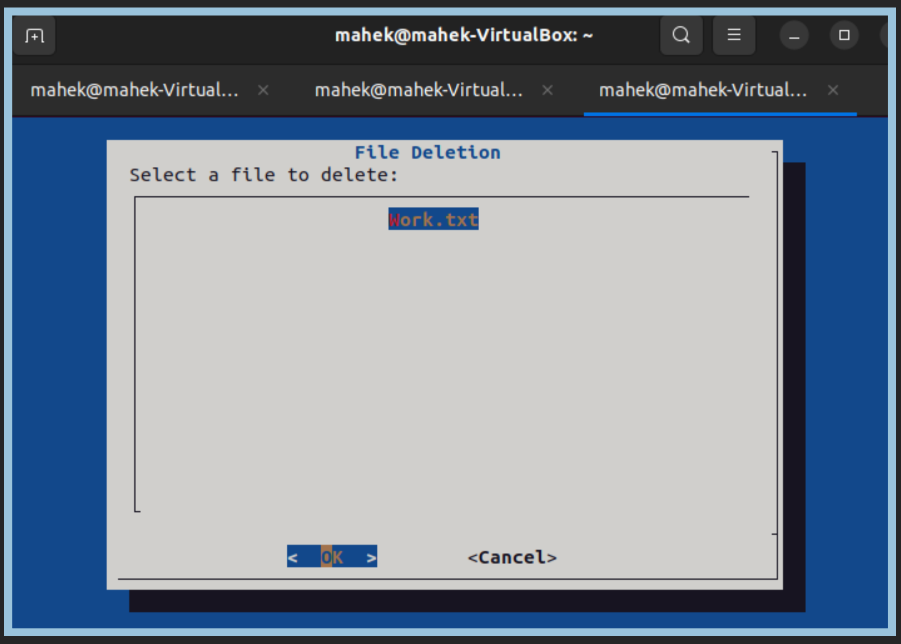


Output Part 1

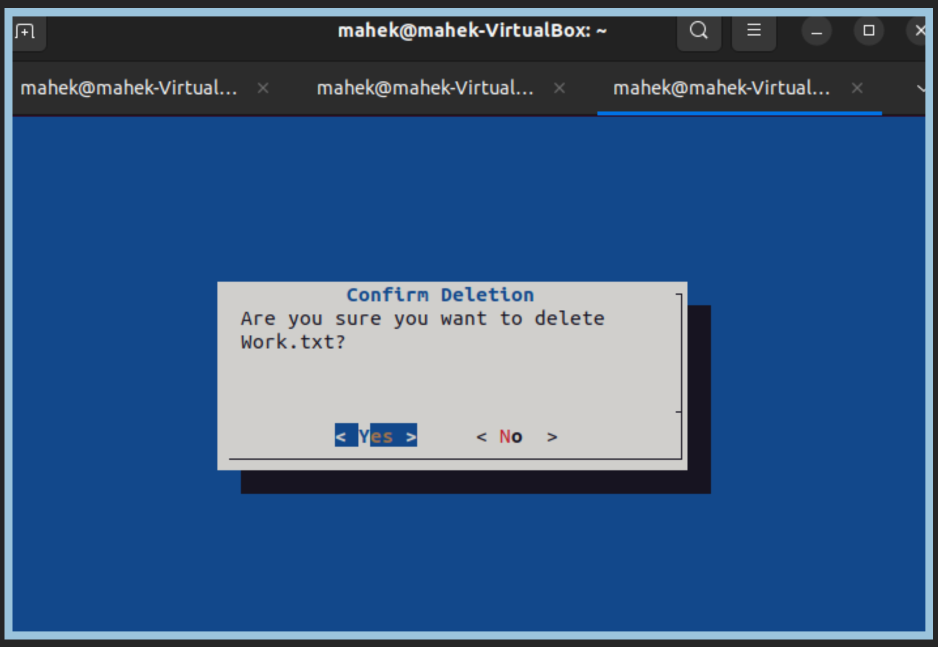
continued...



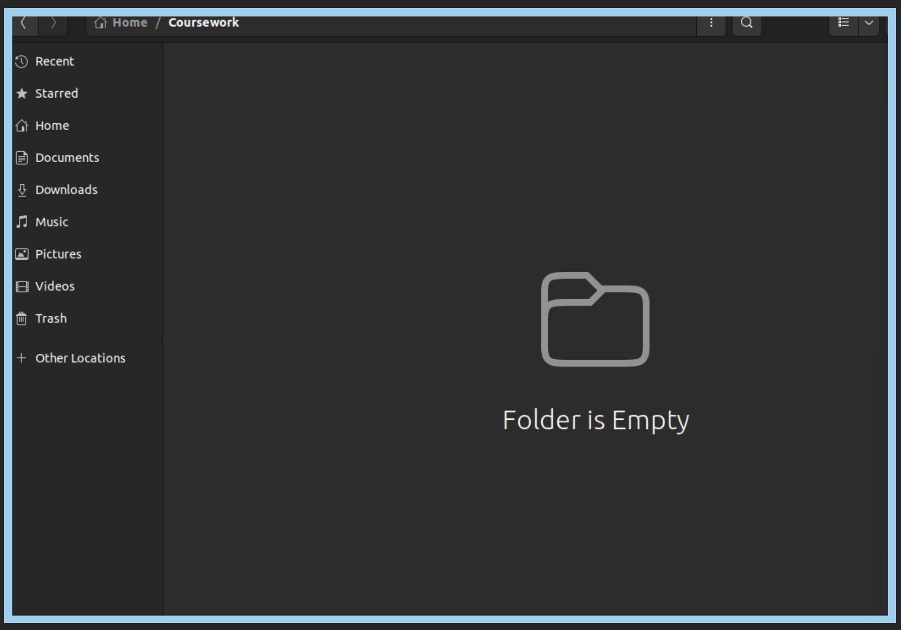
```
mahek@mahek-VirtualBox: ~
mahek@mahek-Virtual...   mahek@mahek-Virtual...   mahek@mahek-Virtual...

            Directory Input
  Enter directory path (Press ENTER
  for current directory):

  Coursework

        <  OK  >       <Cancel>
```



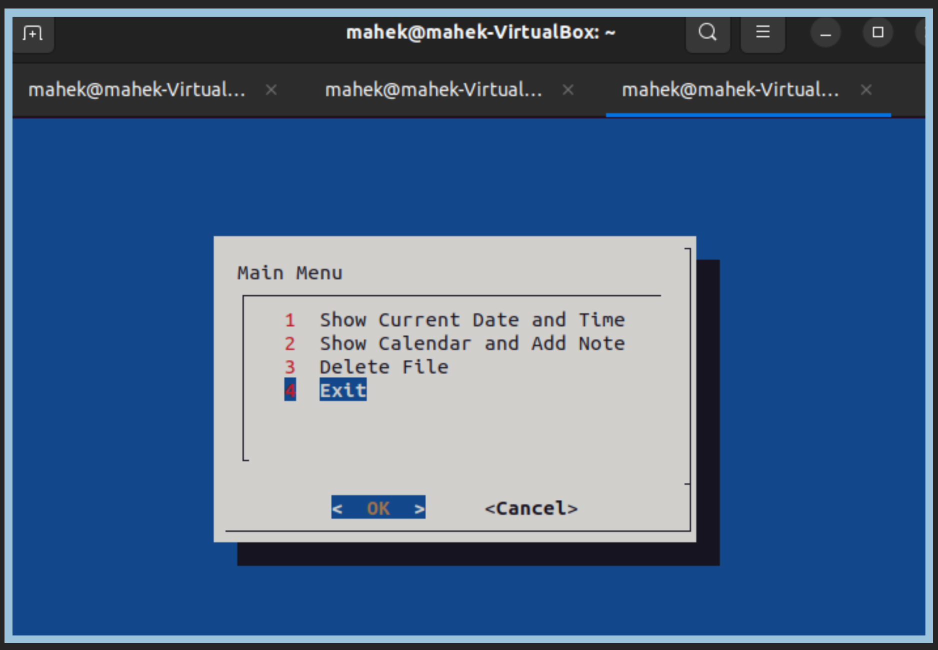
```
mahek@mahek-VirtualBox: ~
mahek@mahek-Virtual...   mahek@mahek-Virtual...   mahek@mahek-Virtual...

                    File Deletion
  Select a file to delete:

                   Work.txt

           <  OK  >          <Cancel>
```



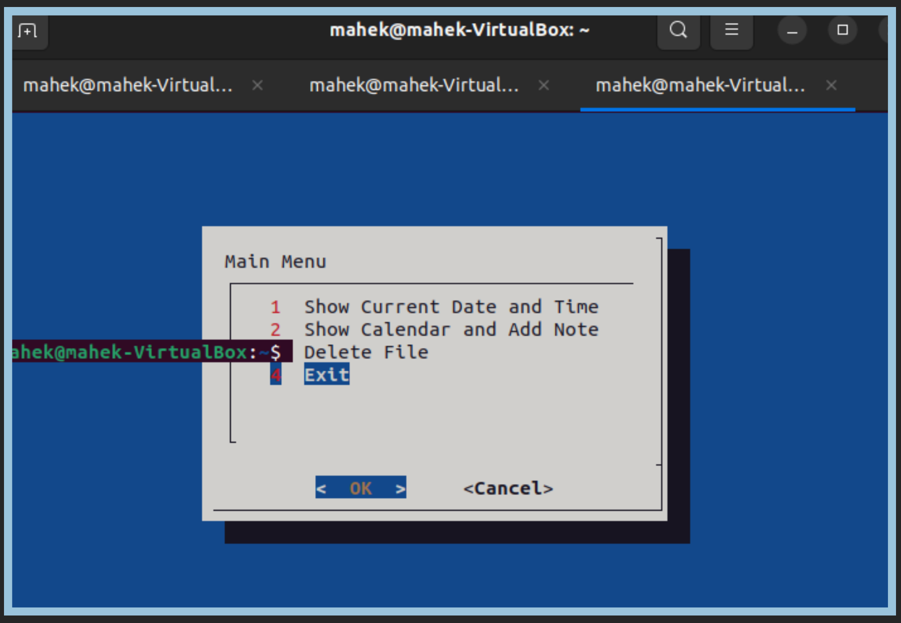
```
mahek@mahek-VirtualBox: ~
mahek@mahek-Virtual...   mahek@mahek-Virtual...   mahek@mahek-Virtual...

            Confirm Deletion
  Are you sure you want to delete
  Work.txt?

         < Yes >      < No  >
```



```
Home / Coursework

Recent
Starred
Home
Documents
Downloads
Music
Pictures
Videos
Trash
+ Other Locations

                 Folder is Empty
```

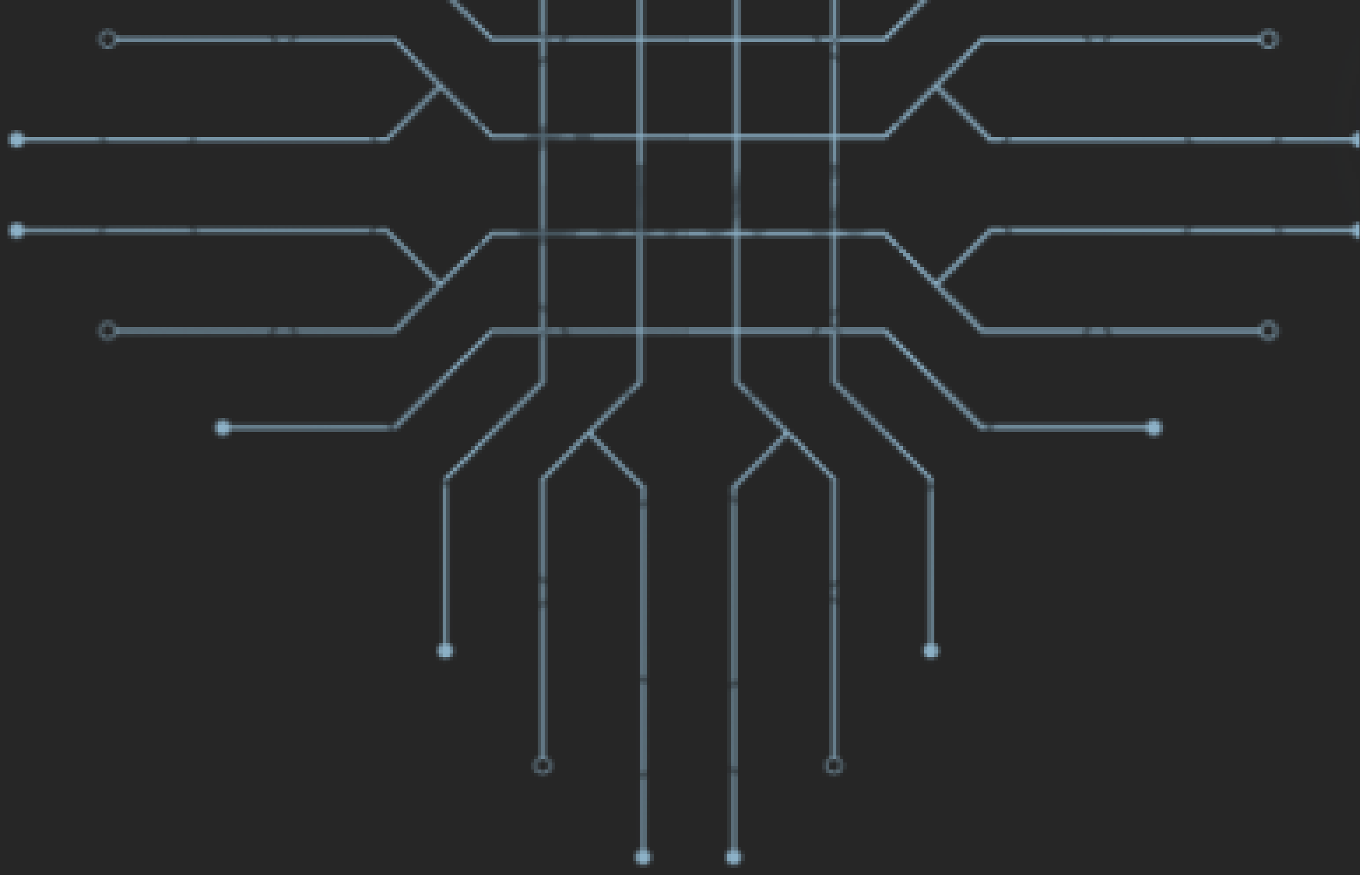


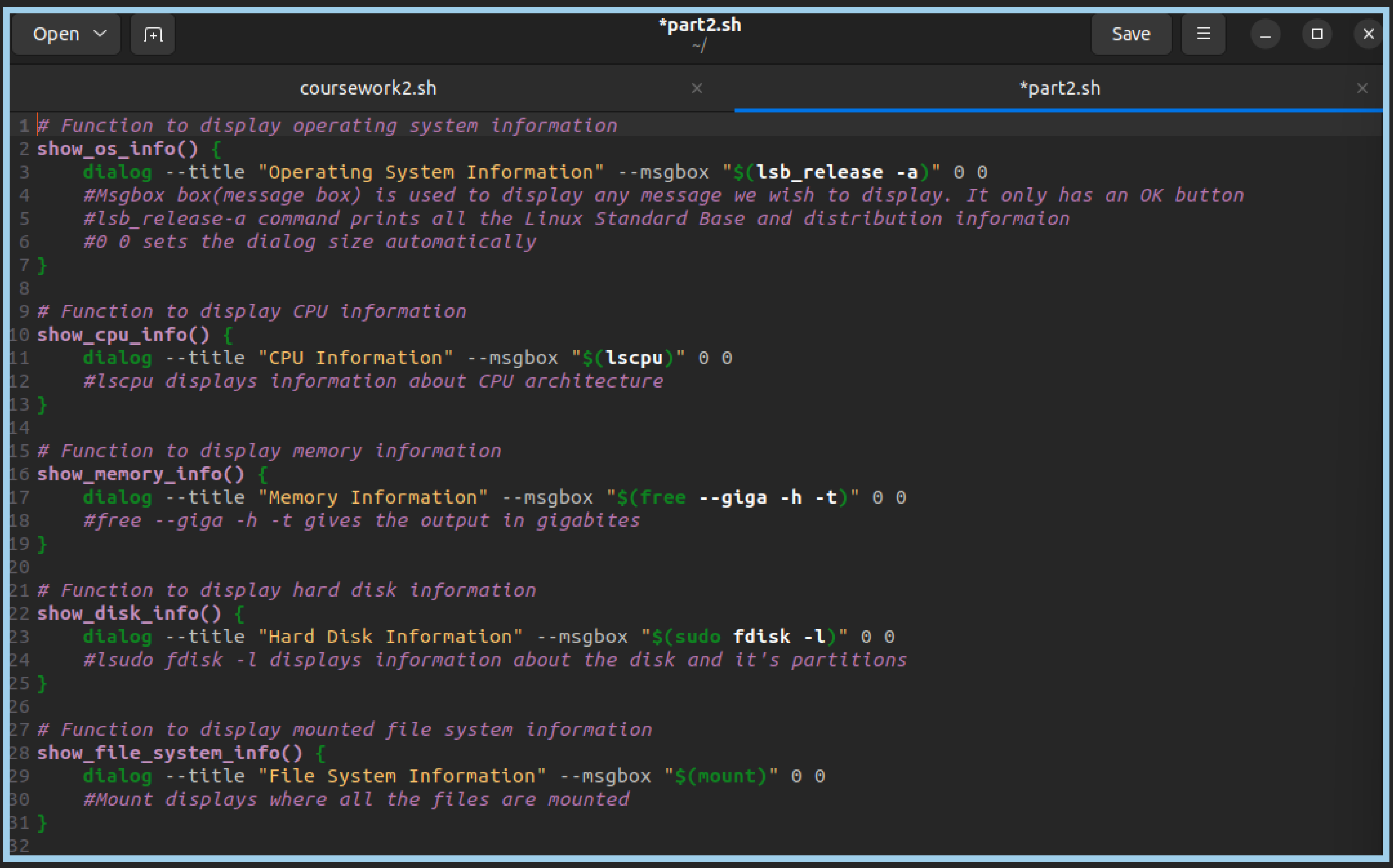
```
mahek@mahek-VirtualBox: ~
mahek@mahek-Virtual...   mahek@mahek-Virtual...   mahek@mahek-Virtual...

  Main Menu

     1   Show Current Date and Time
     2   Show Calendar and Add Note
     3   Delete File
     4   Exit

        <  OK  >       <Cancel>
```



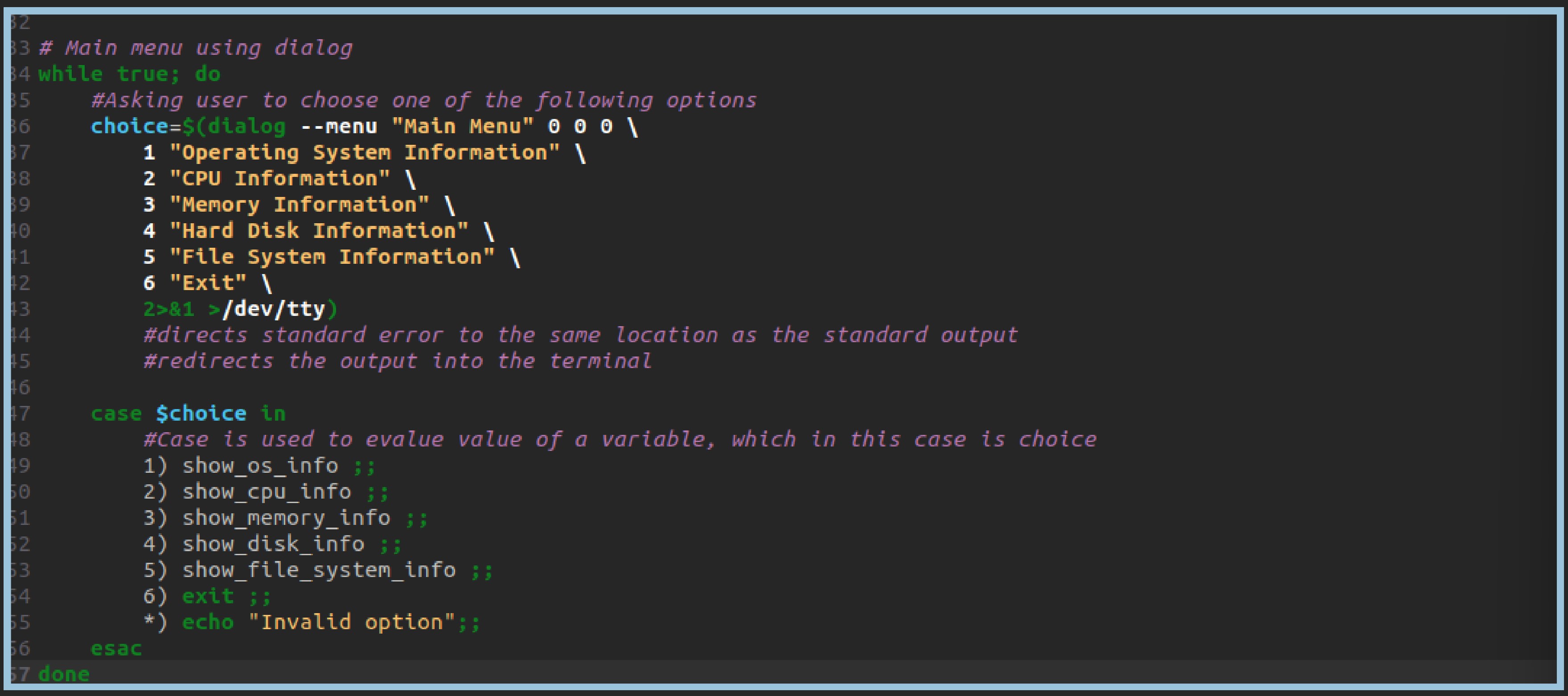
```
mahek@mahek-VirtualBox: ~
mahek@mahek-Virtual...   mahek@mahek-Virtual...   mahek@mahek-Virtual...

  Main Menu

     1   Show Current Date and Time
     2   Show Calendar and Add Note
ahek@mahek-VirtualBox:~$ Delete File
     4   Exit

        <  OK  >       <Cancel>
```

Code Part 2



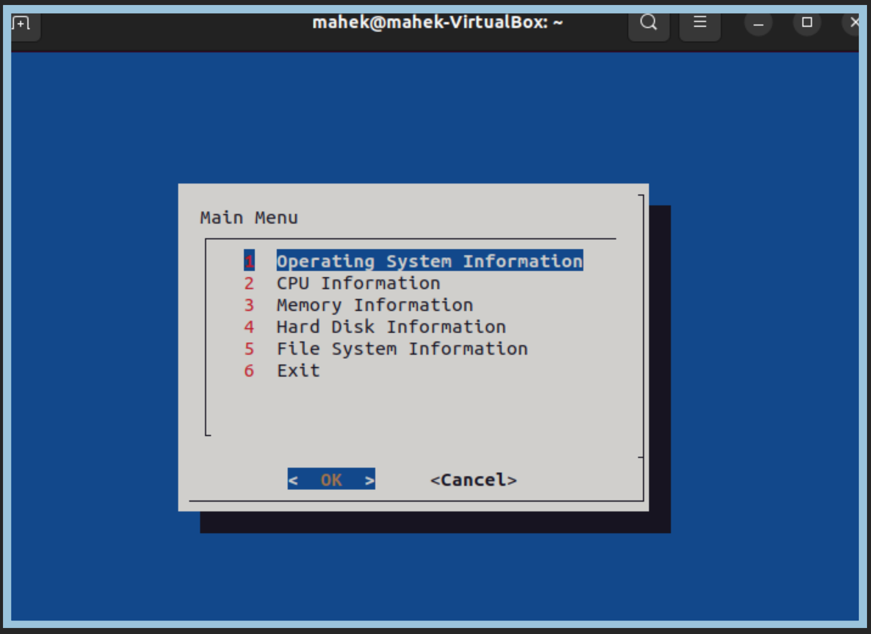


```bash
# Function to display operating system information
show_os_info() {
    dialog --title "Operating System Information" --msgbox "$(lsb_release -a)" 0 0
    #Msgbox box(message box) is used to display any message we wish to display. It only has an OK button
    #lsb_release-a command prints all the Linux Standard Base and distribution informaion
    #0 0 sets the dialog size automatically
}

# Function to display CPU information
show_cpu_info() {
    dialog --title "CPU Information" --msgbox "$(lscpu)" 0 0
    #lscpu displays information about CPU architecture
}

# Function to display memory information
show_memory_info() {
    dialog --title "Memory Information" --msgbox "$(free --giga -h -t)" 0 0
    #free --giga -h -t gives the output in gigabites
}

# Function to display hard disk information
show_disk_info() {
    dialog --title "Hard Disk Information" --msgbox "$(sudo fdisk -l)" 0 0
    #lsudo fdisk -l displays information about the disk and it's partitions
}

# Function to display mounted file system information
show_file_system_info() {
    dialog --title "File System Information" --msgbox "$(mount)" 0 0
    #Mount displays where all the files are mounted
}

```

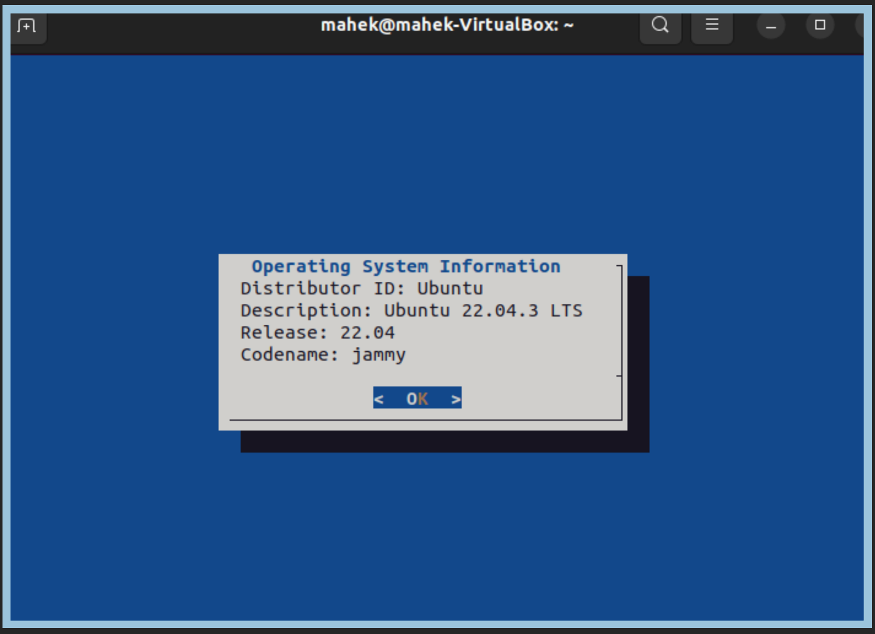


```bash

# Main menu using dialog
while true; do
    #Asking user to choose one of the following options
    choice=$(dialog --menu "Main Menu" 0 0 0 \
        1 "Operating System Information" \
        2 "CPU Information" \
        3 "Memory Information" \
        4 "Hard Disk Information" \
        5 "File System Information" \
        6 "Exit" \
        2>&1 >/dev/tty)
        #directs standard error to the same location as the standard output
        #redirects the output into the terminal

    case $choice in
        #Case is used to evalue value of a variable, which in this case is choice
        1) show_os_info ;;
        2) show_cpu_info ;;
        3) show_memory_info ;;
        4) show_disk_info ;;
        5) show_file_system_info ;;
        6) exit ;;
        *) echo "Invalid option";;
    esac
done
```

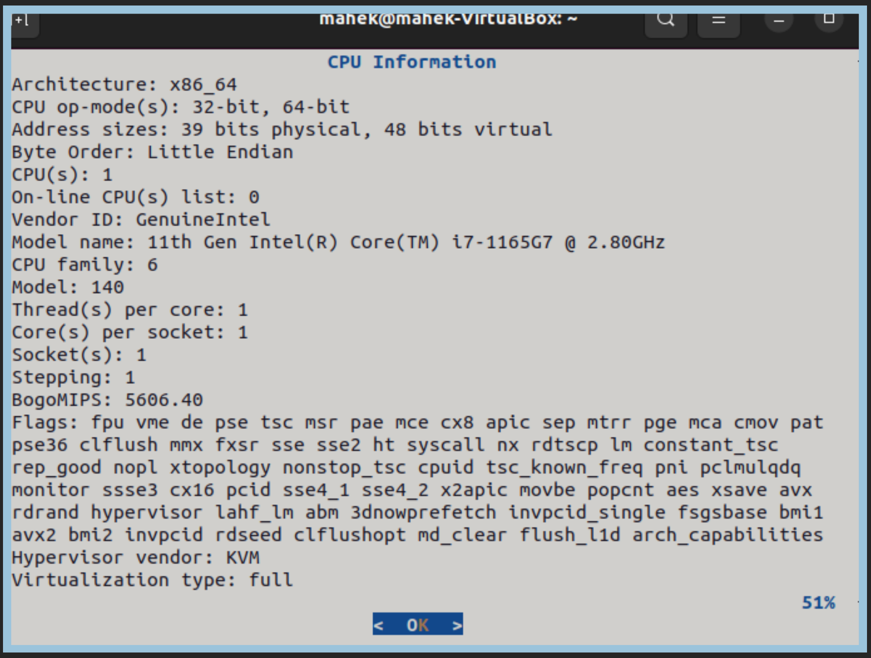
Output part 2



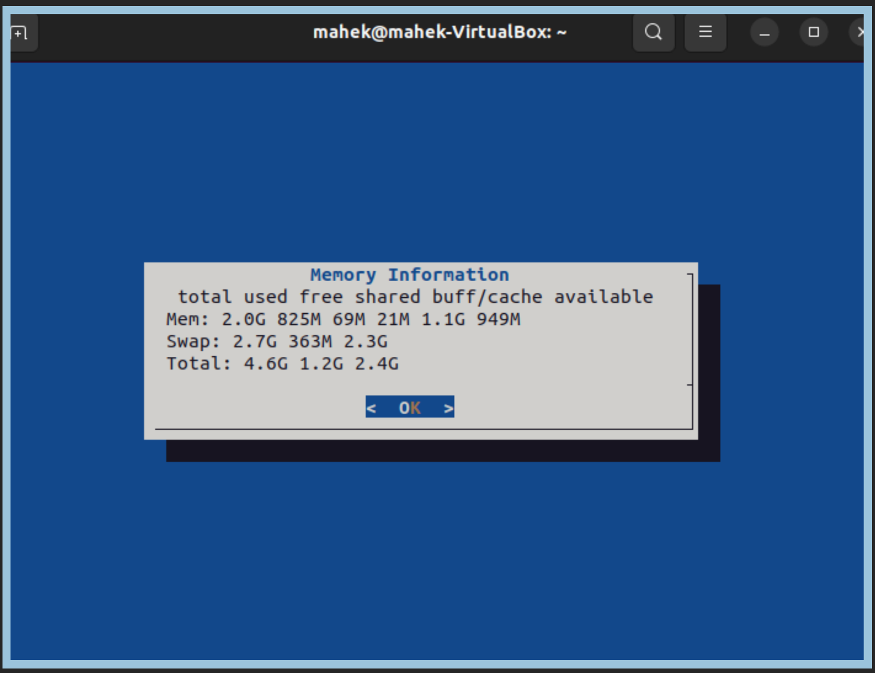
```
mahek@mahek-VirtualBox: ~

Main Menu

  1  Operating System Information
  2  CPU Information
  3  Memory Information
  4  Hard Disk Information
  5  File System Information
  6  Exit

      <  OK  >       <Cancel>
```



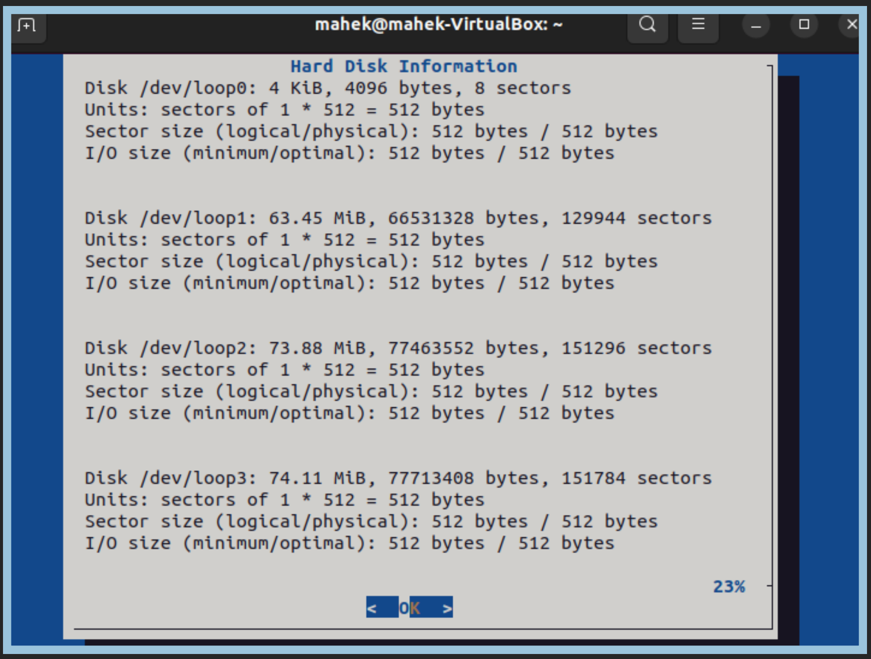
```
mahek@mahek-VirtualBox: ~

Operating System Information
Distributor ID: Ubuntu
Description: Ubuntu 22.04.3 LTS
Release: 22.04
Codename: jammy

      <  OK  >
```



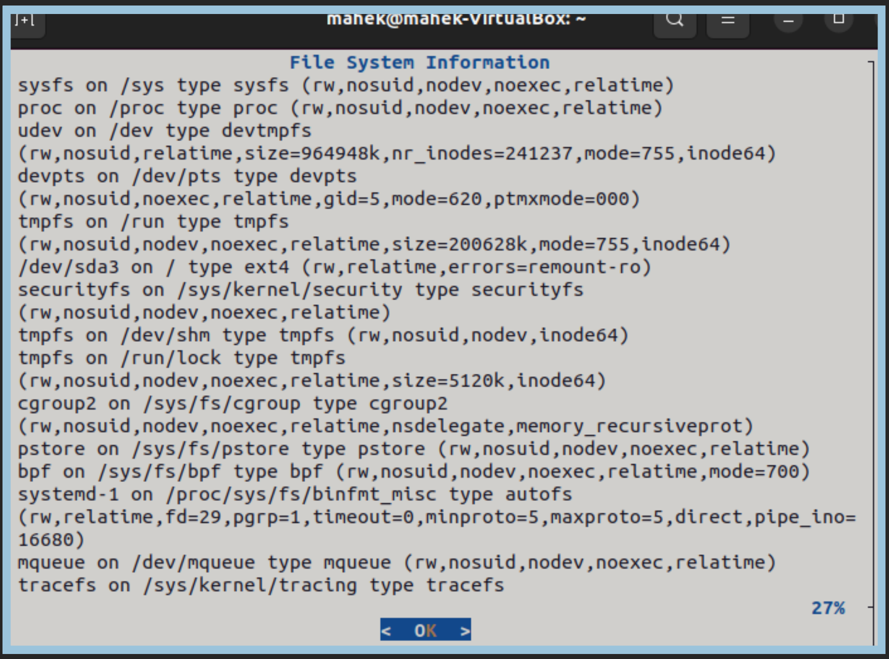
```
mahek@mahek-VirtualBox: ~
                           CPU Information
Architecture: x86_64
CPU op-mode(s): 32-bit, 64-bit
Address sizes: 39 bits physical, 48 bits virtual
Byte Order: Little Endian
CPU(s): 1
On-line CPU(s) list: 0
Vendor ID: GenuineIntel
Model name: 11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz
CPU family: 6
Model: 140
Thread(s) per core: 1
Core(s) per socket: 1
Socket(s): 1
Stepping: 1
BogoMIPS: 5606.40
Flags: fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat
pse36 clflush mmx fxsr sse sse2 ht syscall nx rdtscp lm constant_tsc
rep_good nopl xtopology nonstop_tsc cpuid tsc_known_freq pni pclmulqdq
monitor ssse3 cx16 pcid sse4_1 sse4_2 x2apic movbe popcnt aes xsave avx
rdrand hypervisor lahf_lm abm 3dnowprefetch invpcid_single fsgsbase bmi1
avx2 bmi2 invpcid rdseed clflushopt md_clear flush_l1d arch_capabilities
Hypervisor vendor: KVM
Virtualization type: full
                                                                     51%
                              <  OK  >
```



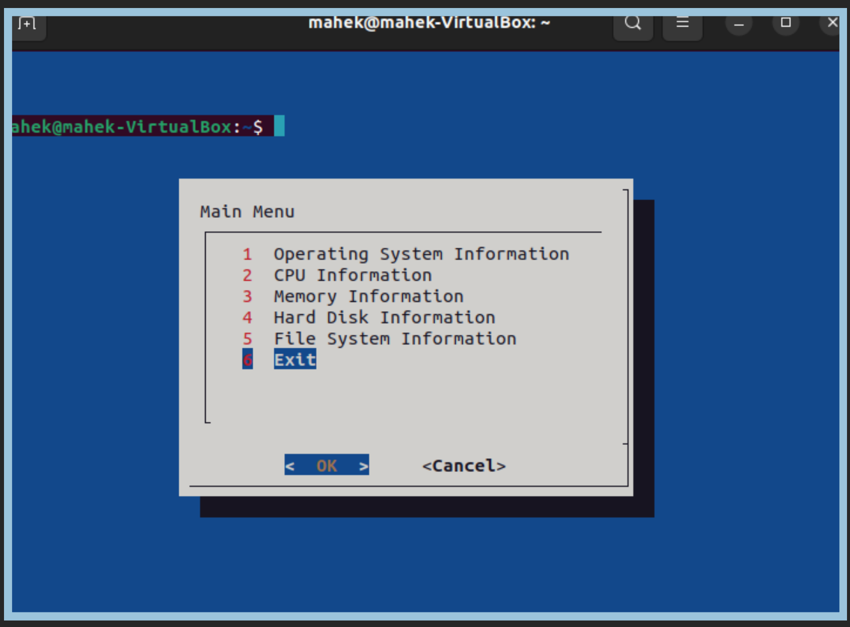
```
mahek@mahek-VirtualBox: ~

            Memory Information
 total used free shared buff/cache available
Mem: 2.0G 825M 69M 21M 1.1G 949M
Swap: 2.7G 363M 2.3G
Total: 4.6G 1.2G 2.4G

              <  OK  >
```



```
mahek@mahek-VirtualBox: ~
        Hard Disk Information
Disk /dev/loop0: 4 KiB, 4096 bytes, 8 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/loop1: 63.45 MiB, 66531328 bytes, 129944 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/loop2: 73.88 MiB, 77463552 bytes, 151296 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/loop3: 74.11 MiB, 77713408 bytes, 151784 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
                                                        23%
                      <  OK  >
```



```
mahek@mahek-VirtualBox: ~
                  File System Information
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs
(rw,nosuid,relatime,size=964948k,nr_inodes=241237,mode=755,inode64)
devpts on /dev/pts type devpts
(rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs
(rw,nosuid,nodev,noexec,relatime,size=200628k,mode=755,inode64)
/dev/sda3 on / type ext4 (rw,relatime,errors=remount-ro)
securityfs on /sys/kernel/security type securityfs
(rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev,inode64)
tmpfs on /run/lock type tmpfs
(rw,nosuid,nodev,noexec,relatime,size=5120k,inode64)
cgroup2 on /sys/fs/cgroup type cgroup2
(rw,nosuid,nodev,noexec,relatime,nsdelegate,memory_recursiveprot)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
bpf on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs
(rw,relatime,fd=29,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=
16680)
mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime)
tracefs on /sys/kernel/tracing type tracefs
                                                                  27%
                           <  OK  >
```



```
mahek@mahek-VirtualBox: ~

ahek@mahek-VirtualBox:~$

Main Menu

  1  Operating System Information
  2  CPU Information
  3  Memory Information
  4  Hard Disk Information
  5  File System Information
  6  Exit

      <  OK  >       <Cancel>
```

Bibliography

- www.cyberciti.biz
- ioflood.com- Linux command blogs
- https://devconnected.com/
- https://stackoverflow.com/
- CST1500 Lab Manual

Reflection

This coursework has been instrumental in facilitating a comprehensive review of the fundamental concepts surrounding Linux bash scripting. Through its execution, we found ourselves delving deeper into the intricate nuances of this principle, allowing us to gain an understanding beyond the theoretical realm. Leveraging the foundational knowledge acquired from our Bash scripting Lab classes, we not only applied the basics but also delved further into advanced techniques, broadening our programming expertise. The process was not solely confined to what was covered in class; it extended to our self-driven research endeavors, enriching our understanding and proficiency in handling complex algorithms and logical structures

THANK YOU