OS ASSIGNMENT 05

NAME : **R NAVEEN KUMAR**

SRN: **PES1UG21CS367**

ROLL NO : **29**

‘**F**’ SECTION

**QUESTION:** Write a C program to truncate the files in a directory created before a certain date to half its original size. Inputs to the program: directory and date as run time arguments.

**CODE:**

*#include <stdio.h>*

*#include <stdlib.h>*

*#include <dirent.h>*

*#include <sys/types.h>*

*#include <sys/stat.h>*

*#include <fcntl.h>*

*#include <unistd.h>*

*#include <time.h>*

*int isCreationTimeBefore(const char \*filePath, const char \*dateTime)*

*{*

*struct stat st;*

*// Get file status*

*if (stat(filePath, &st) == 0)*

*{*

*// Extract the file creation time from the file status structure*

*time\_t creationTime = st.st\_ctime;*

*// Convert the creation time to a struct tm*

*struct tm \*tmCreationTime = gmtime(&creationTime);*

*// Parse the input date time string manually*

*int year, month, day, hour, minute, second;*

*sscanf(dateTime, "%d-%d-%d %d:%d:%d", &year, &month, &day, &hour, &minute, &second);*

*// Set the fields of the struct tm for the input date time*

*struct tm tmDateTime;*

*tmDateTime.tm\_year = year - 1900;*

*tmDateTime.tm\_mon = month - 1;*

*tmDateTime.tm\_mday = day;*

*tmDateTime.tm\_hour = hour;*

*tmDateTime.tm\_min = minute;*

*tmDateTime.tm\_sec = second;*

*// Convert the input date time to time\_t*

*time\_t timeDateTime = mktime(&tmDateTime);*

*// Compare the file creation time with the input date time*

*if (mktime(tmCreationTime) < timeDateTime)*

*{*

*return 1;*

*}*

*else*

*{*

*return 0;*

*}*

*}*

*else*

*{*

*printf("Failed to get file status: %s\n", filePath);*

*return -1;*

*}*

*}*

*void truncateFiles(const char \*dirPath, const char \*datetime)*

*{*

*DIR \*dir;*

*struct dirent \*entry;*

*struct stat st;*

*char filePath[1024];*

*off\_t fileSize;*

*// Open the directory*

*dir = opendir(dirPath);*

*if (dir == NULL)*

*{*

*printf("Failed to open directory: %s\n", dirPath);*

*return;*

*}*

*// Read entries from the directory*

*while ((entry = readdir(dir)) != NULL)*

*{*

*if (entry->d\_type == DT\_REG)*

*{ // Regular file*

*snprintf(filePath, sizeof(filePath), "%s/%s", dirPath, entry->d\_name);*

*// Get file status*

*if (stat(filePath, &st) == 0)*

*{*

*fileSize = st.st\_size;*

*// Truncate file to half its size*

*off\_t newFileSize = fileSize / 2;*

*int fd = open(filePath, O\_WRONLY);*

*if (fd == -1)*

*{*

*printf("Failed to open file: %s\n", filePath);*

*continue;*

*}*

*if (isCreationTimeBefore(filePath, datetime))*

*{*

*if (ftruncate(fd, newFileSize) == -1)*

*{*

*printf("Failed to truncate file: %s\n", filePath);*

*close(fd);*

*continue;*

*}*

*close(fd);*

*printf("Truncated file: %s\n", filePath);*

*}*

*}*

*else*

*{*

*printf("Failed to get file status: %s\n", filePath);*

*}*

*}*

*}*

*// Close the directory*

*closedir(dir);*

*}*

*int main(int argc, char \*argv[])*

*{*

*// Check for correct number of arguments*

*if (argc != 3)*

*{*

*printf("Usage: %s <directory> <date\_time>\n", argv[0]);*

*printf("Example: %s /path/to/directory \"2023-04-26 12∶20∶56\"\n", argv[0]);*

*return 1;*

*}*

*const char \*dirPath = argv[1];*

*const char \*dateTime = argv[2];*

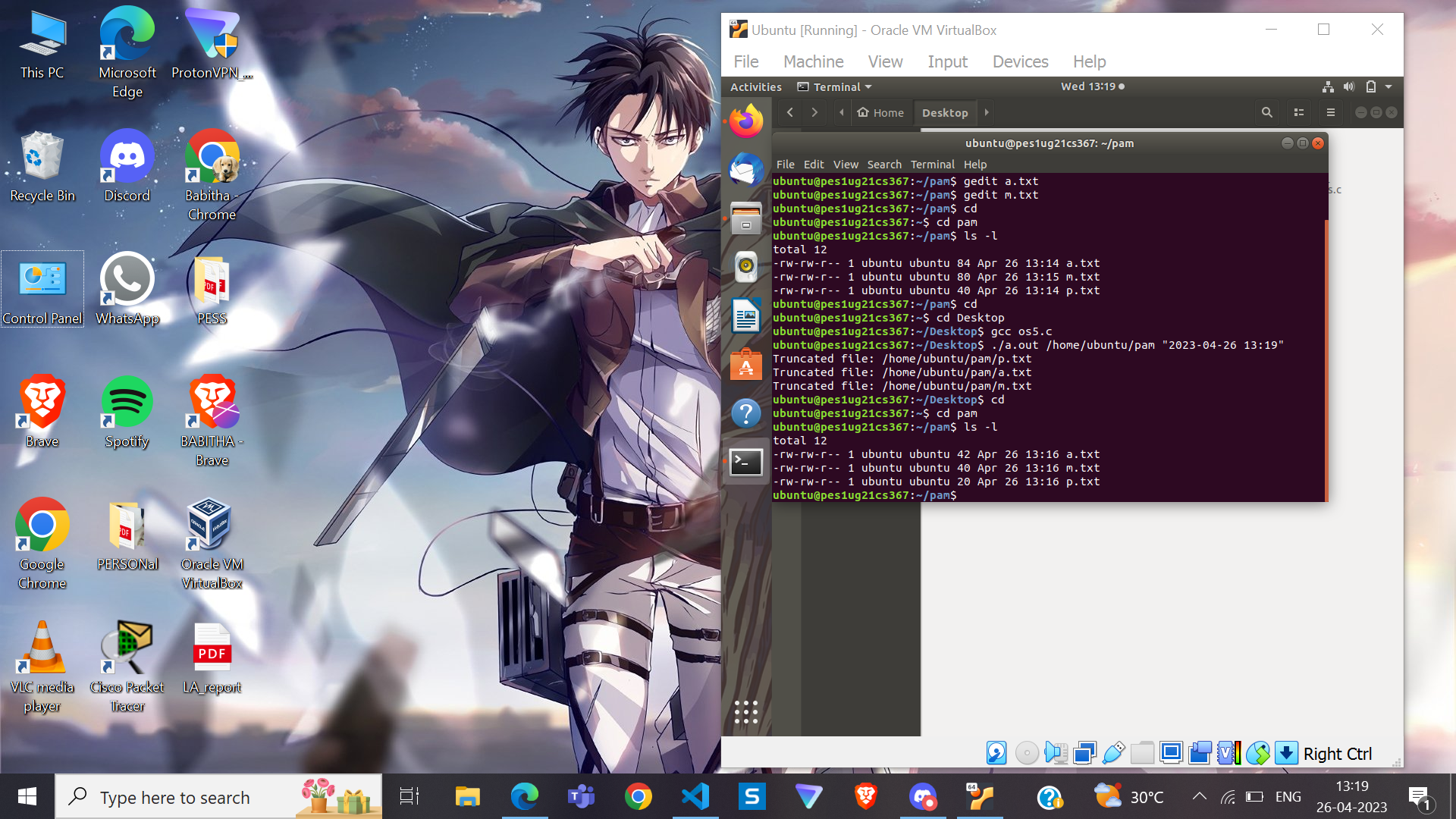
*// Call the truncateFiles function*

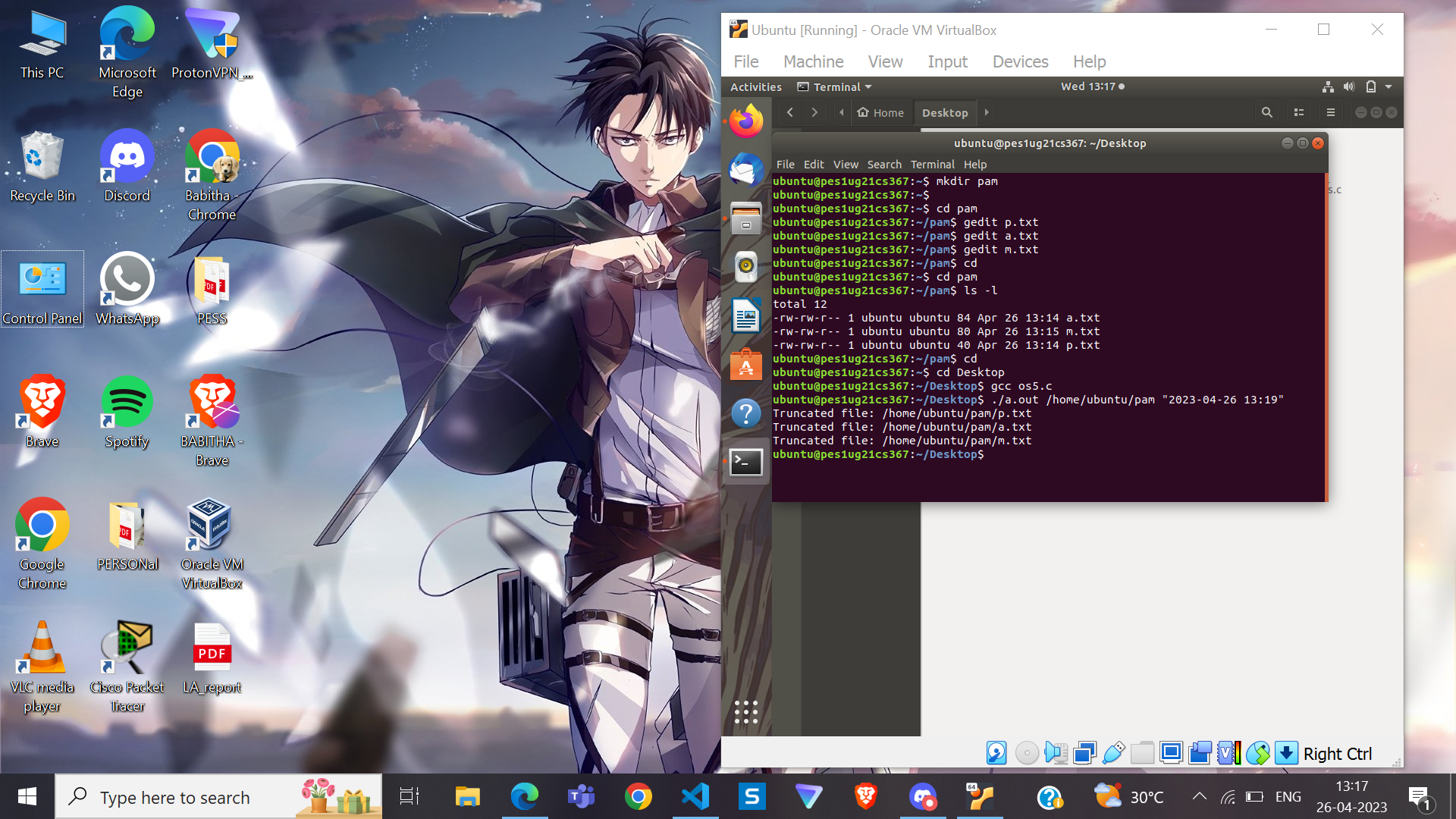
*truncateFiles(dirPath, dateTime);*

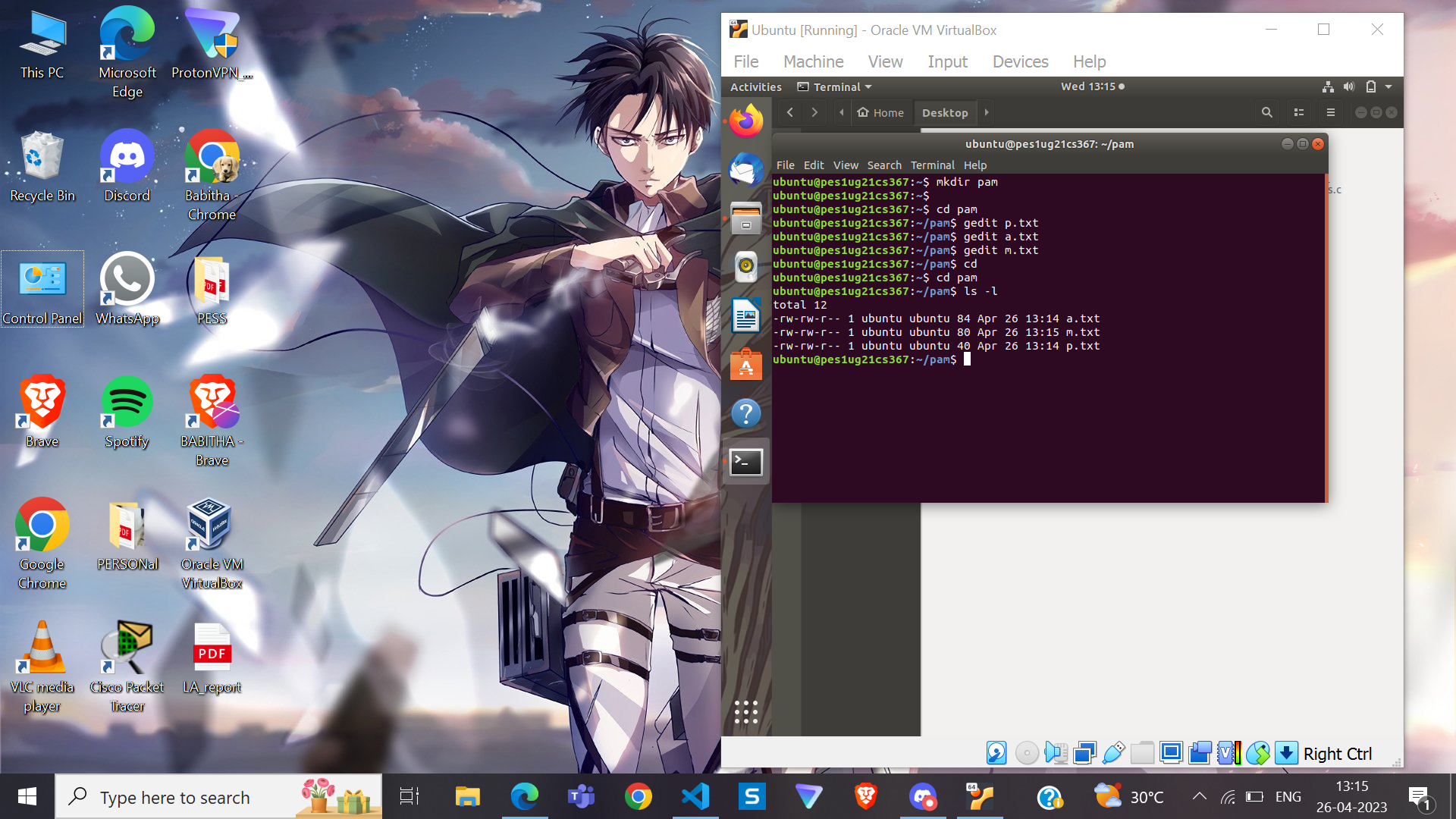
*return 0;*

*}*

**SNAPS:**

****

****

****

---**THANK YOU** ---