OPERATING SYSTEMS

UE22CS242B

PROGRAMMING-EXERCISE-4

4th Semester, Academic Year 2023-2024

Date:18-04-2024

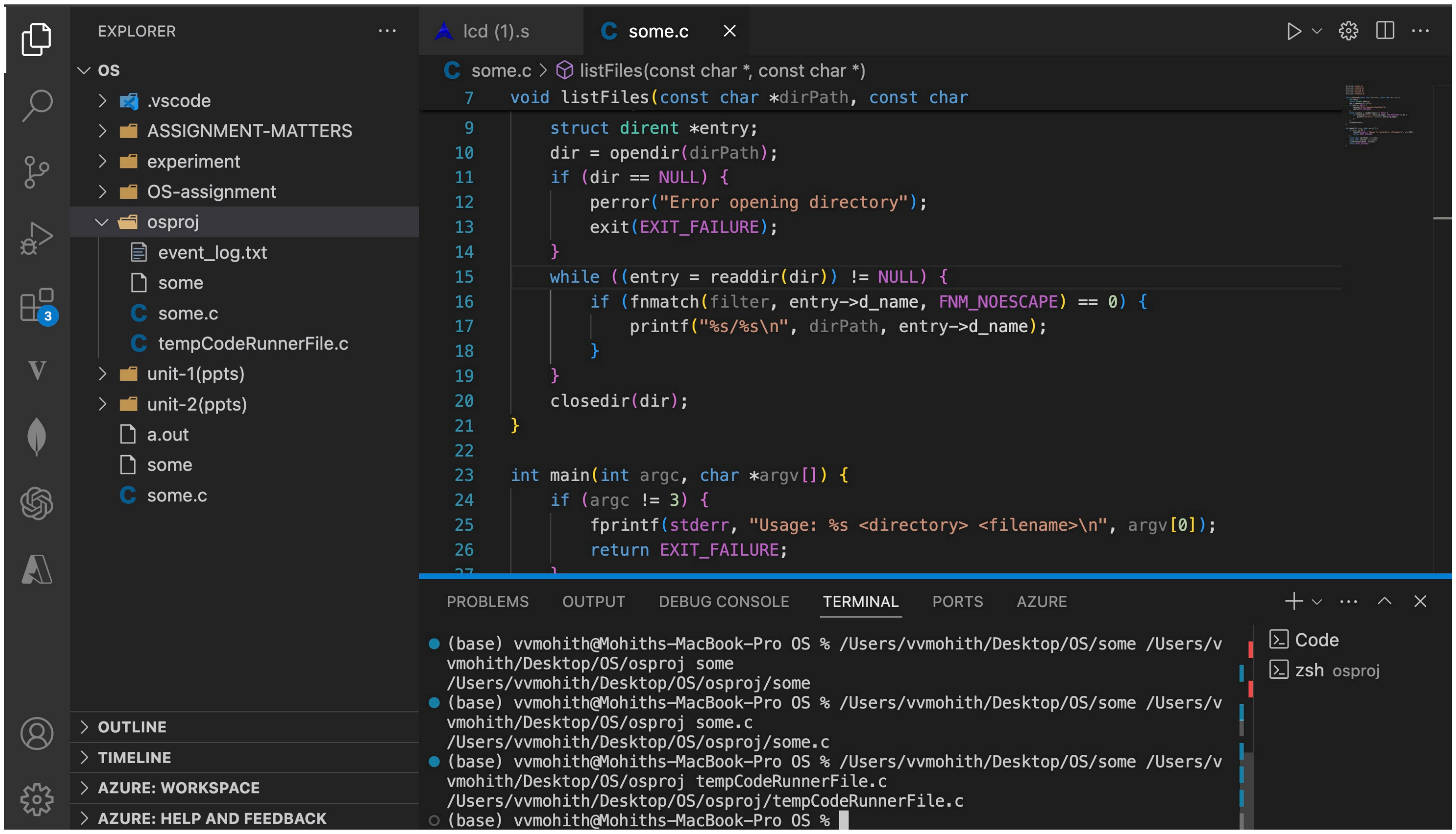
Name: V V Mohith	SRN:-PES2UG22CS641	Section:-K

1) Write a C program to list all files whose name matches the filter.Inputs to the program as run time arguments: directory and filename (need to support wildcard) Example: a.out /home/Ubuntu/ abc1.txt Example: a.out /home/Ubuntu/abc*.txt

CODE:-

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <dirent.h>
#include <fnmatch.h>

void listFiles(const char *dirPath, const char *filter) {
    DIR *dir;
    struct dirent *entry;

    dir = opendir(dirPath);
    if (dir == NULL) {
        perror("Error opening directory");
        exit(EXIT_FAILURE);
    }
    while ((entry = readdir(dir)) != NULL) {
        if (fnmatch(filter, entry->d_name, FNM_NOESCAPE) == 0) {
            printf("%s/%s\n", dirPath, entry->d_name);
        }
    }
    closedir(dir);
}

int main(int argc, char *argv[]) {
    if (argc != 3) {
        fprintf(stderr, "Usage: %s <directory> <filename>\n", argv[0]);
        return EXIT_FAILURE;
    }
    const char *dirPath = argv[1];
    const char *filter = argv[2];
    listFiles(dirPath, filter);
    return EXIT_SUCCESS;
}
```

OUTPUT:-



```c
void listFiles(const char *dirPath, const char
    struct dirent *entry;
    dir = opendir(dirPath);
    if (dir == NULL) {
        perror("Error opening directory");
        exit(EXIT_FAILURE);
    }
    while ((entry = readdir(dir)) != NULL) {
        if (fnmatch(filter, entry->d_name, FNM_NOESCAPE) == 0) {
            printf("%s/%s\n", dirPath, entry->d_name);
        }
    }
    closedir(dir);
}

int main(int argc, char *argv[]) {
    if (argc != 3) {
        fprintf(stderr, "Usage: %s <directory> <filename>\n", argv[0]);
        return EXIT_FAILURE;
```

```
(base) vvmohith@Mohiths-MacBook-Pro OS % /Users/vvmohith/Desktop/OS/some /Users/v
vmohith/Desktop/OS/osproj some
/Users/vvmohith/Desktop/OS/osproj/some
(base) vvmohith@Mohiths-MacBook-Pro OS % /Users/vvmohith/Desktop/OS/some /Users/v
vmohith/Desktop/OS/osproj some.c
/Users/vvmohith/Desktop/OS/osproj/some.c
(base) vvmohith@Mohiths-MacBook-Pro OS % /Users/vvmohith/Desktop/OS/some /Users/v
vmohith/Desktop/OS/osproj tempCodeRunnerFile.c
/Users/vvmohith/Desktop/OS/osproj/tempCodeRunnerFile.c
(base) vvmohith@Mohiths-MacBook-Pro OS %
```

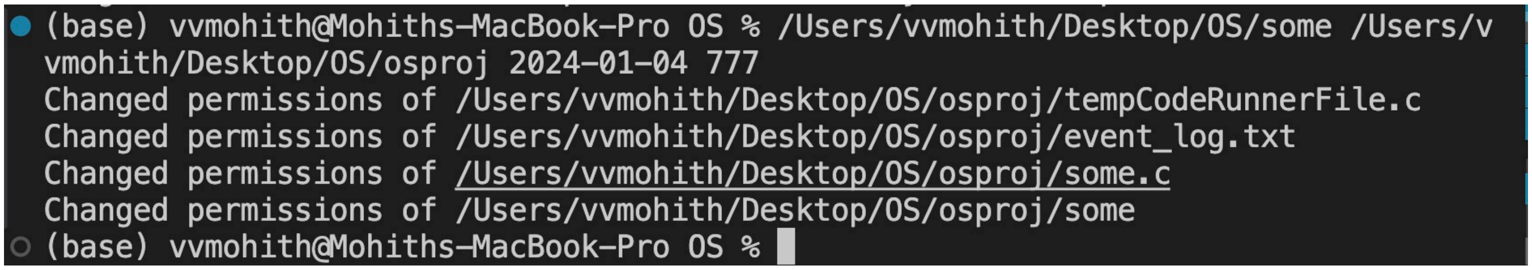
2)2.Write a C program to change the permissions of files in a directory created after a certain date. Inputs to the program: directory/file, date and new permission to be set as run time arguments

CODE:-

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <dirent.h>
#include <string.h>
#include <unistd.h>
#include <time.h>

void changePermissions(const char *dirPath, const char
*dateStr, mode_t newPermissions);

void changePermissionsRecursive(const char *dirPath, const
char *dateStr, mode_t newPermissions);

int main(int argc, char *argv[]) {
    if (argc != 4) {
        fprintf(stderr, "Usage: %s <directory> <date>
<permissions>\n", argv[0]);
        return EXIT_FAILURE;
    }

    const char *dirPath = argv[1];
    const char *dateStr = argv[2];
    mode_t newPermissions = strtol(argv[3], NULL, 8);

    changePermissions(dirPath, dateStr, newPermissions);

    return EXIT_SUCCESS;
}
```

```
void changePermissions(const char *dirPath, const char
*dateStr, mode_t newPermissions) {
    changePermissionsRecursive(dirPath, dateStr,
newPermissions);
}

void changePermissionsRecursive(const char *dirPath, const
char *dateStr, mode_t newPermissions) {
    DIR *dir;
    struct dirent *entry;
    struct stat fileStat;

    if ((dir = opendir(dirPath)) == NULL) {
        perror("Error opening directory");
        return;
    }

    while ((entry = readdir(dir)) != NULL) {
        char path[1024];
        snprintf(path, sizeof(path), "%s/%s", dirPath, entry-
>d_name);

        if (strcmp(entry->d_name, ".") == 0 || strcmp(entry-
>d_name, "..") == 0)
            continue;

        if (lstat(path, &fileStat) < 0) {
            perror("Error stating file");
            continue;
        }

        if (S_ISDIR(fileStat.st_mode)) {
            changePermissionsRecursive(path, dateStr,
newPermissions); // Recurse into subdirectory
        } else {
            // Check file creation date
            time_t creationTime = fileStat.st_ctime;
            struct tm *tm_creation = localtime(&creationTime);
            char creationDateStr[11]; // YYYY-MM-DD
            strftime(creationDateStr, sizeof(creationDateStr), "%Y-
%m-%d", tm_creation);


            // Compare creation date with provided date
            if (strcmp(creationDateStr, dateStr) > 0) {
                // Change file permissions
                if (chmod(path, newPermissions) != 0) {
                    perror("Error changing permissions");
                } else {
                    printf("Changed permissions of %s\n", path);
                }
            }
        }
    }

    closedir(dir);
}
```

OUTPUT:-



```
● (base) vvmohith@Mohiths-MacBook-Pro OS % /Users/vvmohith/Desktop/OS/some /Users/v
  vmohith/Desktop/OS/osproj 2024-01-04 777
  Changed permissions of /Users/vvmohith/Desktop/OS/osproj/tempCodeRunnerFile.c
  Changed permissions of /Users/vvmohith/Desktop/OS/osproj/event_log.txt
  Changed permissions of /Users/vvmohith/Desktop/OS/osproj/some.c
  Changed permissions of /Users/vvmohith/Desktop/OS/osproj/some
○ (base) vvmohith@Mohiths-MacBook-Pro OS %
```

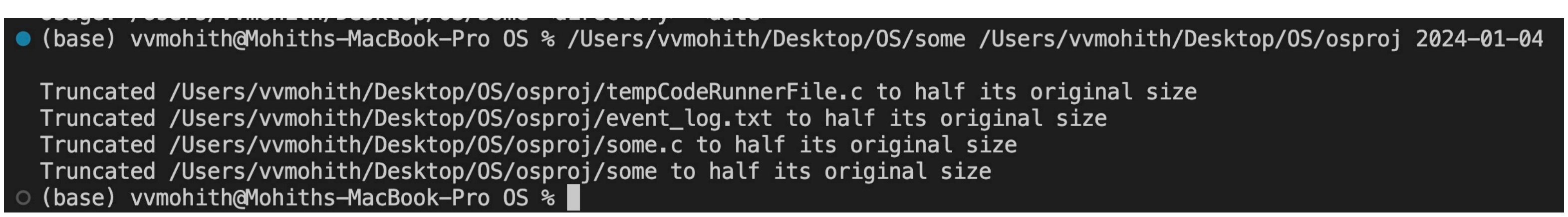
3. Write a C program to truncate the files in a directory created after a certain Date to half its original size. Inputs to the program: directory/file and date as run time arguments

CODE:-

```
#include <stdio.h>

#include <stdlib.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <dirent.h>

#include <string.h>

#include <unistd.h>

#include <time.h>


void truncateFiles(const char *dirPath, const char *dateStr);


void truncateFilesRecursive(const char *dirPath, const char
*dateStr);


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

    if (argc != 3) {
        fprintf(stderr, "Usage: %s <directory> <date>\n", argv[0]);
        return EXIT_FAILURE;
    }

    const char *dirPath = argv[1];
    const char *dateStr = argv[2];

    truncateFiles(dirPath, dateStr);

    return EXIT_SUCCESS;
}

void truncateFiles(const char *dirPath, const char *dateStr) {
    truncateFilesRecursive(dirPath, dateStr);
}

void truncateFilesRecursive(const char *dirPath, const char
*dateStr) {
    DIR *dir;
    struct dirent *entry;
    struct stat fileStat;
    if ((dir = opendir(dirPath)) == NULL) {
        perror("Error opening directory");
        return;
    }

    while ((entry = readdir(dir)) != NULL) {
        char path[1024];
        snprintf(path, sizeof(path), "%s/%s", dirPath, entry-
>d_name);

        if (strcmp(entry->d_name, ".") == 0 || strcmp(entry-
>d_name, "..") == 0)
            continue;

        if (lstat(path, &fileStat) < 0) {
            perror("Error stating file");
            continue;
        }

        if (S_ISDIR(fileStat.st_mode)) {
            truncateFilesRecursive(path, dateStr); // Recurse into
subdirectory
        } else {
            // Check file creation date
            time_t creationTime = fileStat.st_ctime;
            struct tm *tm_creation = localtime(&creationTime);
            char creationDateStr[11]; // YYYY-MM-DD
            strftime(creationDateStr, sizeof(creationDateStr), "%Y-
%m-%d", tm_creation);

            // Compare creation date with provided date
            if (strcmp(creationDateStr, dateStr) > 0) {
                // Truncate file to half its original size
                off_t originalSize = fileStat.st_size;
                off_t newSize = originalSize / 2;

                if (truncate(path, newSize) != 0) {
                    perror("Error truncating file");
                } else {
                    printf("Truncated %s to half its original size\n",
path);
```

```
                }
            }
        }
    }

    closedir(dir);
}
```

OUTPUT:-



```
● (base) vvmohith@Mohiths-MacBook-Pro OS % /Users/vvmohith/Desktop/OS/some /Users/vvmohith/Desktop/OS/osproj 2024-01-04

  Truncated /Users/vvmohith/Desktop/OS/osproj/tempCodeRunnerFile.c to half its original size
  Truncated /Users/vvmohith/Desktop/OS/osproj/event_log.txt to half its original size
  Truncated /Users/vvmohith/Desktop/OS/osproj/some.c to half its original size
  Truncated /Users/vvmohith/Desktop/OS/osproj/some to half its original size
○ (base) vvmohith@Mohiths-MacBook-Pro OS %
```

Disclaimer:

• The programs and output submitted is duly written, verified and executed by me.

• I have not copied from any of my peers nor from the external resource such as internet.

• If found plagiarized, I will abide with the disciplinary action of the University.

Signature:

[illegible]

Name:V V Mohith
SRN:PES2UG22CS641
Section: K
Date:22-03-2024