System Programming

T.C.
SAKARYA UNIVERSITY
COMPUTER AND IT SCIENCE FACULITY
SOFTWARE ENGINEERING DEPARTMENT
2022 / 2023 FALL SEMESTER
SWE 203 – SYSTEM PROGRAMMING COURSE PROJECT

STUDENT NUMBER – NAME SURNAME
Sadık Mert Dinçel B221202372 sadik.dincel@ogr.sakarya.edu.tr
Oğuzhan Fatih Küçük B221202373 oguzhan.kucuk@ogr.sakarya.edu.tr
Birol Şahin B221202350 birol.sahin2@ogr.sakarya.edu.tr

LECTURER OF THE COURSE
Doç.Dr. Ünal ÇAVUŞOĞLU

SWE 301 – SYSTEM PROGRAMMING COURSE PROJECT

This project was created to combine files with txt extension containing certain data. Our project is basically based on 2 purposes: file merging and splitting. It allows us to combine files and archive them in another txt file, and then recreate the files wherever we want using the archived file.

The codes and outputs of our project's requirements are given below.

First section: Merging Files

- **Input files can only be text files (ASCII, 1 byte per character)**

```c
void checkFilesTxtOrNot(int argc, char * argv[]) {// This function controls if input files include ".txt"
                            //so that we can control whether they are
                            //txt files or not.
    for (i = 2; i < argc - 2; i++) {
        if (strstr(argv[i], ".txt") == NULL) {
            printf("%s input file format is not compatible.\n", argv[i]);
            exit(1);
        }
```

```
    }
}
```

```
fatih@fatih:~/MasaüstÜ$ tarsau -b file1.txt file2.txt file3.txt file.pdf -o c.sau
file.pdf input file format is not compatible.
fatih@fatih:~/MasaüstÜ$
```

- **The name of the archive file must be given after the –o parameter.**

```
 outputfileName = argv[argc - 1];   // This takes the output file name as argv.
```

- **If the archive file name is not given, it should be a.sau by default.**

```
char * outputfileName = "a.sau"; // Default output file name.
```

- **The total size of input files cannot exceed 200 MBytes.**

```
#define MAX_FILE_SIZE 209715200  // 200 MBytes for checking the max value of outputFile.

void checkMaxSize(int totalSize) {           // This function controls if totalsize is bigger than 200mb.
    if (totalSize > MAX_FILE_SIZE) {
        printf("Total Size is bigger than 200MB, which is the top limit\n");
        exit(1);
    }
}
```

- **The number of input files cannot be more than 32.**

```
#define MAX_NUM_FILES 32         // The maximum number of input files will be checked.

void checkMaxFile(int argc) { // If argc is more than 36, which tells there are more than 32 input files.

    if (argc > MAX_NUM_FILES + 4) { //4 comes other parameters
        printf("You are not allowed to merge more than 32 files\n");
        exit(1);
    }
}
```

```
fatih@fatih:~/MasaüstÜ$ tarsau -b file1.txt file2.txt file3.txt file3.txtfile1.txt file2
.txt file3.txt file3.txtfile1.txt file2.txt file3.txt file3.txtfile1.txt file2.txt file3
.txt file3.txtfile1.txt file2.txt file3.txt file3.txtfile1.txt file2.txt file3.txt file3
.txtfile1.txt file2.txt file3.txt file3.txtfile1.txt file2.txt file3.txt file3.txtfile1.
txt file2.txt file3.txt file3.txtfile1.txt file2.txt file3.txt file3.txtfile1.txt file2.
txt file3.txt file3.txtfile1.txt file2.txt file3.txt file3.txtfile1.txt file2.txt file3.
txt file3.txtfile1.txt file2.txt file3.txt file3.txt -o c.sau
You are not allowed to merge more than 32 files
```

Second section: Separating Files

- **It should take at most 2 parameters after -a.**

```
if (argc != 4) { // Check if user enter 4 parameter, 2 parameter after -a
            printf("Invalid number of arguments.\n");
            exit(1);
        }
```

```
fatih@fatih:~/MasaüstÜ$ tarsau -a c.sau d2 test1
Invalid number of arguments.
fatih@fatih:~/MasaüstÜ$
```

- **The first parameter after -a must be the name of the archive file (*.sau). When an inappropriate file name is entered: "Archive file is inappropriate or corrupt!" should write the message.**

```
// We are checking if the archive file name is appropriate
        if (strstr(archiveFileName, ".sau") == NULL) {
            printf("Archive file is inappropriate or corrupt.\n");
            exit(1);
        }
```

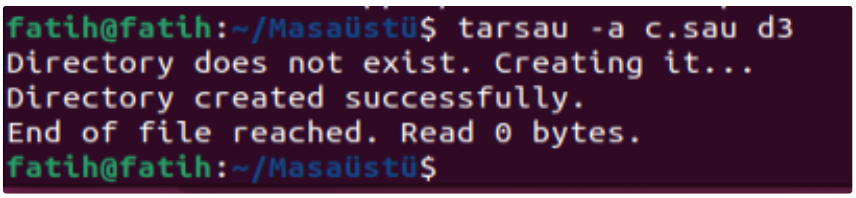
```
fatih@fatih:~/MasaüstÜ$ tarsau -a abc.txt  d2
Archive file is inappropriate or corrupt.
fatih@fatih:~/MasaüstÜ$
```

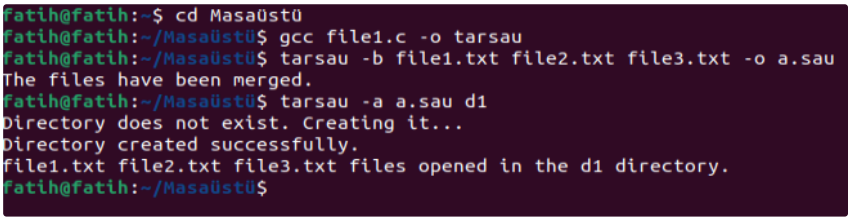
- **The second parameter after -a is the directory name parameter.**

```
const char * directoryPath = argv[3];   //This part takes the directory of files will be opened.
```

- **If there is no place in the entered directory name, it is first created and then files are placed in it.**

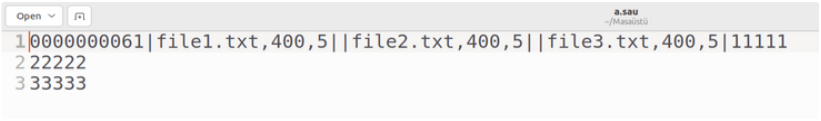
```
struct stat st;
        if (stat(directoryPath, & st) == 0) {
            if (S_ISDIR(st.st_mode)) {
                //printf("Directory exists.\n");
            } else {
                printf("A file with the same name exists.\n");
            }
        } else {
            // Directory does not exist
            printf("Directory does not exist. Creating it...\n");

            // Create the directory
            if (mkdir(directoryPath, 0777) == 0) {
                printf("Directory created successfully.\n");
            } else {
                perror("Error creating directory");
                return 1;
            }
        }
```





-- sau archive file format:

- **The first 10 bytes hold the numerical size of the first section in ASCII.**
- **Each subsequent record '|' It is separated by.**
- **The fields within the record are separated by commas and are: |File name, permissions, size|.**
- **Archive files start immediately after the last recording ends.**



This picture represent to executing the project with using makefile.

This is the github repository that we have used in the course of project:https://github.com/phoenix9897/SystemPrograming/tree/main