УО «Белорусский государственный университет информатики и радиоэлектроники»

Кафедра ПОИТ

Отчет по лабораторной работе № 7

по предмету «Архитектура компьютерной техники и операционных систем»

Вариант 6

Выполнил:

Дедов Н.Ю.

гр. 251003

Проверил:

Аврамец Д.В.

Минск 2023

**1.Исходный код программы в VS code**:

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <dirent.h>

#include <sys/stat.h>

#include <sys/wait.h>

#include <unistd.h>

#include <fcntl.h>

#define DT\_DIR 4

#define DT\_REG 8

void searchInDirectory(const char \*directory, int minSize, int maxSize, const char \*resultFile, int maxProcesses) {

int count = 0;

DIR \*dir;

struct dirent \*entry;

if ((dir = opendir(directory)) == NULL) {

perror("opendir");

return;

}

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

if (entry->d\_type == DT\_DIR && strcmp(entry->d\_name, ".") != 0 && strcmp(entry->d\_name, "..") != 0) {

if (fork() == 0) {

char newDirectory[1024];

snprintf(newDirectory, sizeof(newDirectory), "%s/%s", directory, entry->d\_name);

searchInDirectory(newDirectory, minSize, maxSize, resultFile, maxProcesses);

exit(0);

}

} else if (entry->d\_type == DT\_REG) {

struct stat statbuf;

char filePath[1024];

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

stat(filePath, &statbuf);

if (S\_IXUSR & statbuf.st\_mode && statbuf.st\_size >= minSize && statbuf.st\_size <= maxSize) {

FILE \*out = fopen(resultFile, "a");

if (out == NULL) {

perror("fopen");

return;

}

fprintf(out, "PID: %d, Path: %s, Name: %s, Size: %lld\n", getpid(), filePath, entry->d\_name, (long long)statbuf.st\_size);

fclose(out);

count++;

}

}

}

while (wait(NULL) > 0)

;

printf("Total files in directory %s: %d\n", directory, count);

closedir(dir);

}

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

if (argc < 5) {

fprintf(stderr, "Usage: %s <minSize> <maxSize> <directory> <resultFile> <maxProcesses>\n", argv[0]);

return 1;

}

int minSize = atoi(argv[1]);

int maxSize = atoi(argv[2]);

const char \*directory = argv[3];

const char \*resultFile = argv[4];

int maxProcesses = atoi(argv[5]);

searchInDirectory(directory, minSize, maxSize, resultFile, maxProcesses);

return 0;

}

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <string.h>**

**#include <dirent.h>**

**#include <sys/stat.h>**

**#include <pthread.h>**

**#define DT\_DIR 4**

**#define DT\_REG 8**

**typedef struct {**

**const char \*directory;**

**int minSize;**

**int maxSize;**

**const char \*resultFile;**

**int \*count;**

**} ThreadArgs;**

**void\* searchInDirectory(void \*args) {**

**ThreadArgs \*threadArgs = (ThreadArgs\*)args;**

**int count = 0;**

**DIR \*dir;**

**struct dirent \*entry;**

**if ((dir = opendir(threadArgs->directory)) == NULL) {**

**perror("opendir");**

**return NULL;**

**}**

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

**if (entry->d\_type == DT\_DIR && strcmp(entry->d\_name, ".") != 0 && strcmp(entry->d\_name, "..") != 0) {**

**ThreadArgs newThreadArgs;**

**char newDirectory[1024];**

**snprintf(newDirectory, sizeof(newDirectory), "%s/%s", threadArgs->directory, entry->d\_name);**

**newThreadArgs.directory = newDirectory;**

**newThreadArgs.minSize = threadArgs->minSize;**

**newThreadArgs.maxSize = threadArgs->maxSize;**

**newThreadArgs.resultFile = threadArgs->resultFile;**

**newThreadArgs.count = &count;**

**searchInDirectory(&newThreadArgs);**

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

**struct stat statbuf;**

**char filePath[1024];**

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

**stat(filePath, &statbuf);**

**if (S\_IXUSR & statbuf.st\_mode && statbuf.st\_size >= threadArgs->minSize && statbuf.st\_size <= threadArgs->maxSize) {**

**FILE \*out = fopen(threadArgs->resultFile, "a");**

**if (out == NULL) {**

**perror("fopen");**

**return NULL;**

**}**

**fprintf(out, "Path: %s, Name: %s, Size: %lld\n", filePath, entry->d\_name, (long long)statbuf.st\_size);**

**fclose(out);**

**count++;**

**}**

**}**

**}**

**printf("Total files in directory %s: %d\n", threadArgs->directory, count);**

**\*(threadArgs->count) += count;**

**closedir(dir);**

**return NULL;**

**}**

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

**if (argc < 5) {**

**fprintf(stderr, "Usage: %s <minSize> <maxSize> <directory> <resultFile>\n", argv[0]);**

**return 1;**

**}**

**int minSize = atoi(argv[1]);**

**int maxSize = atoi(argv[2]);**

**const char \*directory = argv[3];**

**const char \*resultFile = argv[4];**

**int count = 0;**

**ThreadArgs args;**

**args.directory = directory;**

**args.minSize = minSize;**

**args.maxSize = maxSize;**

**args.resultFile = resultFile;**

**args.count = &count;**

**searchInDirectory(&args);**

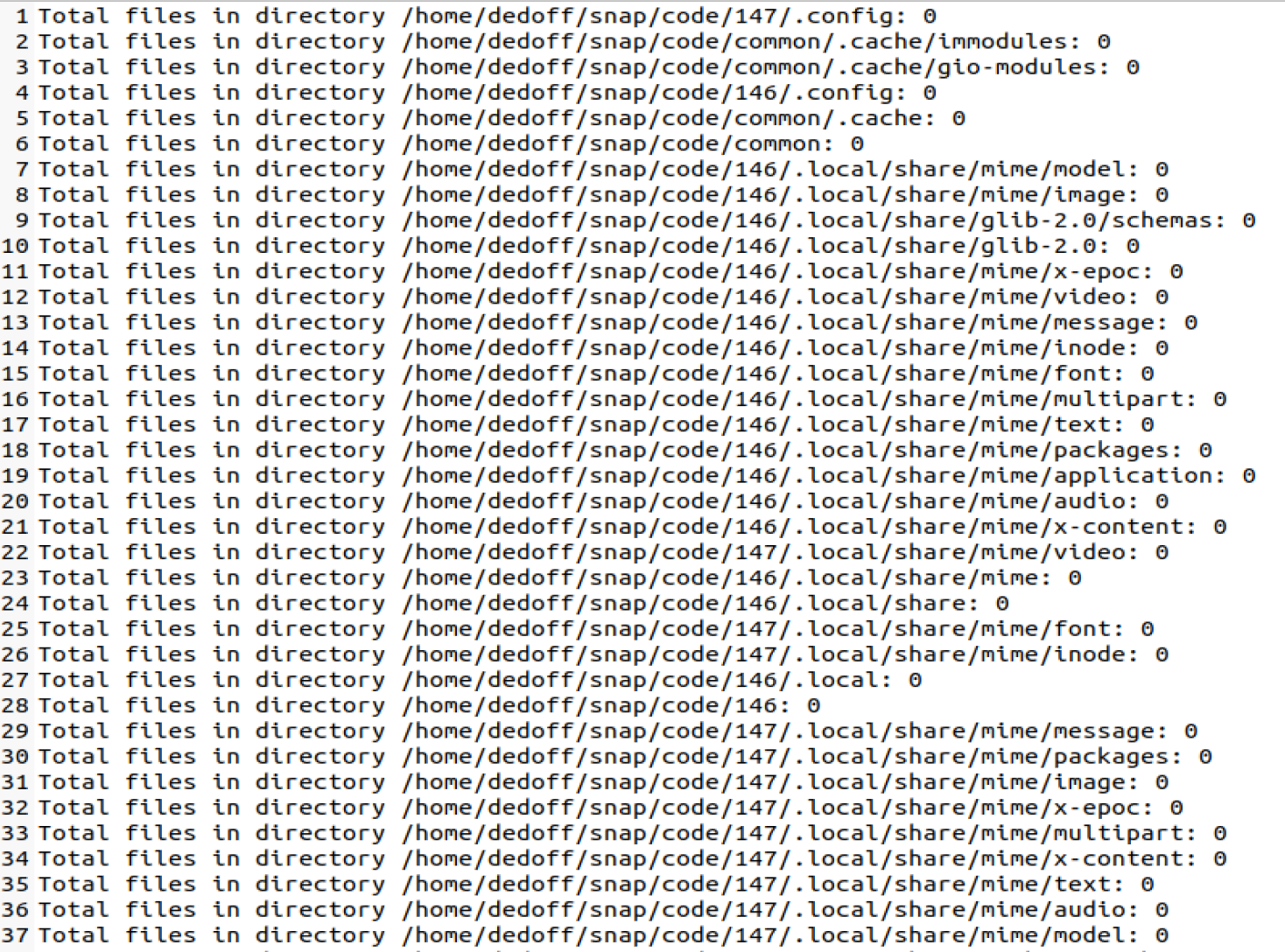
**printf("Total files: %d\n", count);**

**return 0;**

**}**

**2.Результат выполнения программы:**

****

****

**3.Блок-схема алгоритма:**



**Рисунок 1 – Основная схема алгоритма**



**Рисунок 2 – схема алгоритма searchInDirectory (часть 1)**



**Рисунок 3 – схема алгоритма searchInDirectory (часть 2)**