



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Дніпровський національний університет
залізничного транспорту ім. академіка В. Лазаряна

Кафедра КІТ

ЛАБОРАТОРНА РОБОТА № 9
з дисципліни: «Операційні системи»
на тему: «Робота з каталогами і індексними дескрипторами в ОС UNIX»

Виконав: студент групи ПЗ1712

Нікольський О.В.

Приняла: Нежуміра О.І.

Дніпро, 2020

Тема. Робота з каталогами і індексними дескрипторами в ОС UNIX.

Мета роботи:

- вивчити структуру даних, що забезпечують зберігання, пошук по імені, захист і доступ до файлів;
- набути навичок складання програм, що вимагають обробку інформації, що міститься в індексних дескрипторах і каталогах;
- вивчити елементарні прийоми обробки помилок мовою C++ в UNIX.

Опис специфікацій

Task2.cpp

Файл з основною програмою;

Алгоритм роботи:

1. Відкрити директорію, шлях до котрої було вказано при запуску програми;
2. Допоки директорія не пуста, отримати інформацію про файл;
3. Якщо права на файл є лише у володаря файлу, вивести інформацію про файл – ім'я, розмір, ідентифікатор користувача, ідентифікатор групи та права у форматі запису ОС Linux;
4. Якщо цей файл директорія – виконати рекурсивний виклик та перейти до п.2;
5. Якщо усі файли в директорії переглянуті – закінчити поточну ітерацію рекурсії.

```
#include <stdio.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <dirent.h>
#include <cstring>
#include <iostream>

// Fill text string with data of file rights
std::string CheckRights(struct stat &file){
    char access[11];
    (file.st_mode & S_IFDIR)?strcat(access,"d"):strcat(access,"-");
    // Owner
    (file.st_mode & S_IRUSR)?strcat(access,"r"):strcat(access,"-");
    (file.st_mode & S_IWUSR)?strcat(access,"w"):strcat(access,"-");
    (file.st_mode & S_IXUSR)?strcat(access,"x"):strcat(access,"-");
    // Group
    (file.st_mode & S_IRGRP)?strcat(access,"r"):strcat(access,"-");
    (file.st_mode & S_IWGRP)?strcat(access,"w"):strcat(access,"-");
    (file.st_mode & S_IXGRP)?strcat(access,"x"):strcat(access,"-");
    // Other
    (file.st_mode & S_IROTH)?strcat(access,"r"):strcat(access,"-");
    (file.st_mode & S_IWOTH)?strcat(access,"w"):strcat(access,"-");
    (file.st_mode & S_IXOTH)?strcat(access,"x"):strcat(access,"-");
    std::string s_access = access;
```

```

    return s_access;
}

// Recursive bypass file tree
void ShowFilesInDirectory(char* dirName){

    DIR* refDir;
    // Open directory
    if ((refDir = opendir(dirName)) == NULL) {
        perror(dirName);
        return;
    }

    // Structure for keeping data of current directory and file
    struct dirent *dirInfo;
    struct stat fileBuf;

    // While directory isn't empty
    while((dirInfo = readdir(refDir))!=NULL){
        // Getting info about file
        stat(dirInfo->d_name, &fileBuf);

        // Check that this file haven't "." or ".." in name
        if (strcmp(dirInfo->d_name, ".") == 0 || strcmp(dirInfo->d_name, "..") == 0){
            continue;
        }

        // If only current user have rights for read-write - show info about file/directory
        if(!(fileBuf.st_mode & (S_IRGRP|S_IWGRP|S_IROTH|S_IWOTH))){

            // File name - File size - B (bytes) - User ID - Group ID - Rights
            printf("%10s\t%10d B\tUID: %-10d\tGID: %-10d\t", dirInfo->d_name, fileBuf.st_size, fileBuf.st_uid, fileBuf.st_gid);
            std::cout << CheckRights(fileBuf) << std::endl;
        }

        // If directory - start recursive bypass
        if(dirInfo->d_type == DT_DIR){

            // Create full path to file - for right work stat()
            strcat(dirName, "/");
            strcat(dirName, dirInfo->d_name);

            // Start recursive
            ShowFilesInDirectory(dirName);

            // Delete name of last directory from path
            std::string helpPath = dirName;

```

```

std::string helpName = dirInfo->d_name;
std::string::size_type it = helpPath.find(helpName);
if(it != std::string::npos){
    helpPath.erase(it, helpName.length());
}

strcpy(dirName, helpPath.c_str());
}
}

// Close current directory
closedir(refDir);
}

int main(int argc, char* argv[]) {
    ShowFilesInDirectory(argv[1]);
    return 0;
}

```

Результати виконання програми

```

Файл  Правка  Вид  Закладки  Настройка  Справка
g++ -o Task2 Task2.cpp
./Task2 /home/alexey/DIIT/repos/Operations-Systems/LR9
video.mp4      1837848 B   UID: 1000   GID: 1000   -rw-----
Task2.cpp      3503 B     UID: 1000   GID: 1000   -rw-----
[alexey@manjaro-kde LR9]$ ls -l
итого 1852
-rwxr-xr-x 1 alexey alexey 19032 июн 13 15:49 main
-rw-r--r-- 1 alexey alexey 92 июн 13 15:50 Makefile
-rwxr-xr-x 1 alexey alexey 19032 июн 13 15:50 Task2
-rw----- 1 alexey alexey 3503 июн 13 15:49 Task2.cpp
-rw-r--r-- 1 alexey alexey 3562 июн 13 15:49 Task5.cpp
-rw----- 1 alexey alexey 1837848 июн 9 21:25 video.mp4
drwxr-xr-x 2 alexey alexey 4096 июн 12 16:57 vscode
[alexey@manjaro-kde LR9]$ mkdir SomeDir
[alexey@manjaro-kde LR9]$ ls -l
итого 1856
-rwxr-xr-x 1 alexey alexey 19032 июн 13 15:49 main
-rw-r--r-- 1 alexey alexey 92 июн 13 15:50 Makefile
drwxr-xr-x 2 alexey alexey 4096 июн 13 15:50 SomeDir
-rwxr-xr-x 1 alexey alexey 19032 июн 13 15:50 Task2
-rw----- 1 alexey alexey 3503 июн 13 15:49 Task2.cpp
-rw-r--r-- 1 alexey alexey 3562 июн 13 15:49 Task5.cpp
-rw----- 1 alexey alexey 1837848 июн 9 21:25 video.mp4
drwxr-xr-x 2 alexey alexey 4096 июн 12 16:57 vscode
[alexey@manjaro-kde LR9]$ chmod go-rwx SomeDir
[alexey@manjaro-kde LR9]$ ls -l
итого 1856
-rwxr-xr-x 1 alexey alexey 19032 июн 13 15:49 main
-rw-r--r-- 1 alexey alexey 92 июн 13 15:50 Makefile
drwx----- 2 alexey alexey 4096 июн 13 15:50 SomeDir
-rwxr-xr-x 1 alexey alexey 19032 июн 13 15:50 Task2
-rw----- 1 alexey alexey 3503 июн 13 15:49 Task2.cpp
-rw-r--r-- 1 alexey alexey 3562 июн 13 15:49 Task5.cpp
-rw----- 1 alexey alexey 1837848 июн 9 21:25 video.mp4
drwxr-xr-x 2 alexey alexey 4096 июн 12 16:57 vscode
[alexey@manjaro-kde LR9]$ ./Task2 /home/alexey/DIIT/repos/Operations-Systems/LR9
video.mp4      1837848 B   UID: 1000   GID: 1000   -rw-----
SomeDir        4096 B     UID: 1000   GID: 1000   drwx-----
Task2.cpp      3503 B     UID: 1000   GID: 1000   -rw-----
[alexey@manjaro-kde LR9]$

```

Рис.1 Результат виконання програми

На рис.1 продемонстровано роботу програми.

Описання виконаних дій:

1. Компіляція та запуск (виконано за допомогою Makefile);
2. Результат роботи програми — на даному етапі у директорії знаходиться лише два файли, права на читання-запис котрих має лише поточний користувач;
3. Виведення на екран даних про всі файли у директорії з вказанням їх прав (команда ОС Linux — “ls -l”);
4. Створення нової директорії у поточній (команда “mkdir [name]”);
5. Виведення на екран даних про всі файли у директорії з вказанням їх прав;
6. Редагування прав доступу для директорії SomeDir — тепер лише поточний користувач має право на читання, запис та виконання (команда chmod [options][rights][path]);
7. Виведення на екран даних про всі файли у директорії з вказанням їх прав;
8. Виконання програми — після виконання пунктів 4 та 6 програма виведе і теку SomeDir.

Висновки

Вивчили структуру даних, що забезпечують зберігання, пошук по імені, захист і доступ до файлів. Набули навичок складання програм, що вимагають обробку інформації, що міститься в індексних дескрипторах і каталогах. Вивчили елементарні прийоми обробки помилок мовою C++ в UNIX.