ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

###### Московский институт электроники и математики

Лабораторная работа №1 по операционным системам

«Прикладная математика»

группы БПМ-192

Пчелинцев Илья Игоревич

Вариант 17

|  |  |
| --- | --- |
|  |  |
| Москва 2021 | |

**Условие(Вариант 17)**

Написать программу, определяющую права доступа для всех файлов текущего каталога. Передать эту информацию через файловую систему в параллельный процесс и вывести данные только о тех файлах,

дата создания которых не превышает семи дней от текущей даты.

Предусмотреть возможность прерывания от клавиатуры.

**Код программы**

#include <stdio.h>

#include <stdlib.h>

#include <signal.h>

#include <setjmp.h>

#include <unistd.h>

#include <sys/wait.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <time.h>

#include <fcntl.h>

#include <stdint.h>

//Написать программу, определяющую права доступа для всех файлов текущего каталога.

//Передать эту информацию через файловую систему в

//параллельный процесс и вывести данные только о тех файлах,

//дата создания которых не превышает семи дней от текущей даты.

//Предусмотреть возможность прерывания от клавиатуры.

sigjmp\_buf obl; //Область памяти для запоминания состояния процессаi

int intCount = 0; //Счётчик прерываний

void interrupt() //Подпрограмма обработки прерываний

{

intCount++;

if (intCount == 1) //Если количество прерываний = 1

{

printf("\nFirst interruption!\n");

}

else {

printf("\nLast interruption!\n");

exit(0);

}

sleep(2);

siglongjmp(obl, 1);//Переход к сохранённому стеку

}

int main() {

int s;

signal(SIGINT, interrupt);//В случае получения сигнала прерывания SIGINT, управление передаётся функции interrupt

sigsetjmp(obl, 1); //Сохраняет текущее состояние стека

while(1)

{

char\* fname ="interfile.txt";

sigsetjmp(obl, 1);

if (fork() == 0)//Создание межпроцессного канала

{

int f = open(fname, O\_CREAT | O\_TRUNC | O\_WRONLY, 0644); //открываем файл для записи в него информации о каталоге

// O\_CREAT - создать если не существует

// O\_TRUNC - очистить если заполнен

// O\_WRONLY - только на запись

if (f == -1) { //открылся ли файл

printf("Not Ok...!\n");

exit(1);

}

dup2(f, 1);//Дублирование дескриптора межпроцессного канала на стандартный вывод

execlp("ls", "ls", "-og", "--time-style=long-iso", NULL);//Вывод списка файлов в межпроцессный канал

close(f);

exit(1);

}

else

{

wait(&s);//Ожидание окончания дочернего процесса

sigsetjmp(obl, 1);

char rights[15];

int k\_hard\_links;

int size;

int year;

int month;

int day;

int hour;

int min;

char name[50];

struct stat buff;

time\_t now = time(NULL);

FILE\* fd = fopen(fname, "r");

if (fd == NULL) {

printf("NOT Ok\n");

exit(1);

}

char cc[255]; //выделяем память под строку

char\* status = fgets(cc, sizeof(cc), fd); //считываем построчно

status = fgets(cc, sizeof(cc), fd);

while (status) { //считываем построчно

sscanf(cc, "%15s %d %d %4d-%2d-%2d %2d:%2d %30s", rights, &k\_hard\_links, &size, &year,&month,&day,&hour,&min,name); //считываем все данные из файла,

//printf("rights are %s, year is %d, month is %d, day is %d, name is %s\n", rights, year, month, day, name);

stat(name, &buff);

if ((intmax\_t)(now - buff.st\_mtime) < 7 \* 24 \* 60 \* 60) {

printf(cc);

}

status = fgets(cc, sizeof(cc), fd); //считываем следующую строку

}

fclose(fd);

sigsetjmp(obl, 1);

printf("Program ended its work and will abort in 10 seconds\n");

sleep(10);

exit(1);

}

}

return 0;

}

**Пример работы:**

Лабораторная работа выполнена в системе WSL Ubuntu 21.04

Основные используемые системные вызовы:

open

close

exit

fork

wait

exec

dup2

sigsetjmp

siglongjmp

1. **В данной системе выполнение программы с таким условием невозможно, т.к. система не отслеживает дату создания файлов – даже команда stat не выводит эту дату:**

**Изображение выглядит как текст

Автоматически созданное описание**

1. **Посмотрим сначала содержимое каталога в котором будет выполняться задача – исследуемым временем будет время последней модификации**

**Изображение выглядит как текст, электроника, снимок экрана, компьютер

Автоматически созданное описание**

* Дата использования программы совпадает с датой последней модификации lab17 – 7 декабря
* промежуточный файл interfile.txt был создан во время выполнения программы
* в него через параллельный процесс были записана информация о всех файлах в домашнем каталоге (включая созданный interfile.txt) через команду (ls -og):

Изображение выглядит как текст

Автоматически созданное описание

* из файла видно, что в него входят все файлы (включая Epolicy.pdf и Geo\_Is\_2D\_All.txt, которые давно не модифицировались)
* однако в выводе в консоль записаны лишь о тех файлах, у которых дата модификации (взятая с помощью stat для каждого файла записанного в interfile) отличается от текущей не больше чем на 7 дней
* После выполнения основной программы было выведено Program ended its work and will abort in 10 seconds, и через 10 секунд программа прекратила работу

1. **Проверим теперь работу обработчика прерываний**

* Из кода видно, что последнее сохранение стека было выполнено перед выводом “Program ended its work and will abort in 10 seconds”
* вот что произойдёт, если выполнить одно и два прерывания:

**Изображение выглядит как текст

Автоматически созданное описание**

* При одном прерывании было выведено “First Interuption” и предупреждение об окончании работы программы снова вывелось, и ожидание программы началось снова (то есть, программа будет ждать ещё 10 секунд)
* При двух прерываний программа вывела “First Interuption” от первого прерывания и “Last Interuption” для второй и программа прекратила работу раньше

1. **Программа состоит из двух модулей – основной части и обработчика прерываний**

**Основная программа:**

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/wait.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <time.h>

#include <fcntl.h>

#include <stdint.h>

//Написать программу, определяющую права доступа для всех файлов текущего каталога.

//Передать эту информацию через файловую систему в

//параллельный процесс и вывести данные только о тех файлах,

//дата создания которых не превышает семи дней от текущей даты.

//Предусмотреть возможность прерывания от клавиатуры.

int main() {

int s;

while(1)

{

char\* fname ="interfile.txt";

if (fork() == 0)//Создание межпроцессного канала

{

int f = open(fname, O\_CREAT | O\_TRUNC | O\_WRONLY, 0644); //открываем файл для записи в него информации о каталоге

// O\_CREAT - создать если не существует

// O\_TRUNC - очистить если заполнен

// O\_WRONLY - только на запись

if (f == -1) { //открылся ли файл

printf("Not Ok...!\n");

exit(1);

}

dup2(f, 1);//Дублирование дескриптора межпроцессного канала на стандартный вывод

execlp("ls", "ls", "-og", "--time-style=long-iso", NULL);//Вывод списка файлов в межпроцессный канал

close(f);

exit(1);

}

else

{

wait(&s);//Ожидание окончания дочернего процесса

char rights[15];

int k\_hard\_links;

int size;

int year;

int month;

int day;

int hour;

int min;

char name[50];

struct stat buff;

time\_t now = time(NULL);

FILE\* fd = fopen(fname, "r");

if (fd == NULL) {

printf("NOT Ok\n");

exit(1);

}

char cc[255]; //выделяем память под строку

char\* status = fgets(cc, sizeof(cc), fd); //считываем построчно

status = fgets(cc, sizeof(cc), fd);

while (status) { //считываем построчно

sscanf(cc, "%15s %d %d %4d-%2d-%2d %2d:%2d %30s", rights, &k\_hard\_links, &size, &year,&month,&day,&hour,&min,name); //считываем все данные из файла,

//printf("rights are %s, year is %d, month is %d, day is %d, name is %s\n", rights, year, month, day, name);

stat(name, &buff);

if ((intmax\_t)(now - buff.st\_mtime) < 7 \* 24 \* 60 \* 60) {

printf(cc);

}

status = fgets(cc, sizeof(cc), fd); //считываем следующую строку

}

fclose(fd);

printf("Program ended its work and will abort in 10 seconds\n");

sleep(10);

exit(1);

}

}

return 0;

}

**Пример работы:**

**Изображение выглядит как текст, электроника, снимок экрана

Автоматически созданное описание**

**Обработчик прерываний:**

#include <stdio.h>

#include <stdlib.h>

#include <signal.h>

#include <setjmp.h>

#include <unistd.h>

//Написать программу, определяющую права доступа для всех файлов текущего каталога.

//Передать эту информацию через файловую систему в

//параллельный процесс и вывести данные только о тех файлах,

//дата создания которых не превышает семи дней от текущей даты.

//Предусмотреть возможность прерывания от клавиатуры.

sigjmp\_buf obl; //Область памяти для запоминания состояния процессаi

int intCount = 0; //Счётчик прерываний

void interrupt() //Подпрограмма обработки прерываний

{

intCount++;

if (intCount == 1) //Если количество прерываний = 1

{

printf("\nFirst interruption!\n");

}

else {

printf("\nLast interruption!\n");

exit(0);

}

sleep(2);

siglongjmp(obl, 1);//Переход к сохранённому стеку

}

int main() {

signal(SIGINT, interrupt);//В случае получения сигнала прерывания SIGINT, управление передаётся функции interrupt

sigsetjmp(obl, 1); //Сохраняет текущее состояние стека

while(1)

{

sigsetjmp(obl, 1);

printf("Program ended its work and will abort in 10 seconds\n");

sleep(10);

exit(1);

}

return 0;

}

**Пример работы:**

**Изображение выглядит как текст

Автоматически созданное описание**

**(Объединение этих модулей даёт требуемую заданием программу)**