Министерство науки и высшего образования Российской Федерации

Новосибирский государственный технический университет

Управление ресурсами в вычислительных системах

Лабораторная работа №3

Факультет: прикладной математики и информатики

Группа: ПМ-63

Студенты: Кожекин М.В.

Утюганов Д.С.

Преподаватели: Стасышин В. М.

Хайленко Е. А.

Новосибирск

2019

1. Условие (Вариант №9)

Разработать программу, демонстрирующую динамику изменения данных в системе упра вления вводом-выводом ОС UNIX

(таблицы файлов и таблицы открытых файлов процессов). Например, сценарий програм мы может быть следующим:

–открытие процессом-предком стандартных файлов ввода-вывода и четырех пользовате льских файлов для чтения;

–закрытие процессом-предком двух пользовательских файлов;

–процесс-предок порождает процесс, который наследует таблицы файлов и открытых ф айлов процесса-предка и закрывает 1 файл;

–завершается процесс-потомок. После каждого из этапов печатаются таблицы файлов и открытых файлов для обоих процессов.

1. Анализ задачи

Если заданный каталог существует, то рекурсивно обходим все его поддиректории и считаем количество папок и файлов внутри. В конце выводим название каждой поддиректории и число содержащихся в ней файлов и папок.

1. Используемые программные средства

**Функции:**

fd open (char \*filename, int oflags) – открытие файла **filename**, режим **oflags**

fstat(int fd, struct stat \*buf) – получение данных о файле с дескриптором **fd** через **statbuf**

1. Спецификация

Программа находится в папке ~/lab3

Чтобы собрать программу нужно ввести “make lab3”

Чтобы запустить программу, нужно использовать команду “./lab3”

В результате работы программы, будет показана динамика изменения таблиц файлов и таблицы открытых файлов.

1. Исходный код

**Текст программы**

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <string.h>  #include <sys/types.h>  #include <sys/stat.h>  #include <fcntl.h>  #include <unistd.h>  int countP = 0, countC = 0;  typedef struct {  char \*name; // имя файла  int fd; // дескриптор файла  struct stat data; // метаданные  } inodeTable;  inodeTable inTab[7]; // Таблица описателей файлов  typedef struct {  char \*name; // имя файла  int linksCount; // Число ссылок  int mode; // режим чтения / записи  int locPoint; // расположение каретки в файле  } filesTable;  filesTable fTab[7]; // Таблица файлов  typedef struct {  char \*name; // имя файла  int fd; // дескриптор файла  } openedFilesTable;  openedFilesTable ofTabP[7], ofTabC[7]; // Таблица открытых файлов  /\* вывод таблицы файлов и таблицы открытых файлов  @function printTable  @param isParent - выводить таблицы процесса-предка или процесса-потомка  \*/  void printTables(){  printf("Таблица файлов:\n");  printf("name\t\tlinks\tmode\tlocP\tref\n");  for(int i = 3; i < countP; ++i)  printf("%s \t%d\t%d\t%d\t%d\n", fTab[i].name, fTab[i].linksCount, fTab[i].mode, fTab[i].locPoint, &inTab[i]);  printf("\nТаблица открытых файлов:\n");  printf("name\t\tfd\tref\n");  for(int i = 0; i < countP; ++i)  printf("%s \t%d\t%d\n", ofTabP[i].name, ofTabP[i].fd, &inTab[i]);  for(int i = 0; i < countC; ++i)  printf("%s \t%d\t%d\n", ofTabC[i].name, ofTabC[i].fd, &inTab[i]);    printf("\n\n\n");  }  /\* наследование таблиц при вызове fork()  происходит копирование элементов таблицы открытых файлов и  увеличение числа ссылок на файлы на 1  @function copyTables  \*/  void copyTables(){  for (int i = 0; i < countP; ++i){  fTab[i].linksCount++;  ofTabC[i] = ofTabP[i];  }  countC = countP;  }  /\* закрытие файлов дочернего процесса  @function closeChildProcessFiles  \*/  void closeChildProcessFiles(){  for (int i = countC - 1; i>=0; --i)  fTab[i].linksCount--;    countC = 0;  }  int main(int argc, char \*argv[]) {  char \*names[] = {"stdin","stdout","stderr", "file1.txt", "file2.txt", "file3.txt", "file4.txt"};  // Шаг 1  printf("Открытие стандартных файлов ввода-ввывода и 4 пользовательских файлов родительским процессом\n");  for(int i = 0; i < 3; ++i) {  ofTabP[i].name = names[i];  ofTabP[i].fd = i;  }  for(int i = 3; i < 7; ++i) {  int fd = open(names[i], O\_APPEND); // дескриптор файла  inTab[i].name = names[i];  inTab[i].fd = fd;  fstat(fd, &inTab[i].data);  fTab[i].name = names[i];  fTab[i].linksCount = 1;  fTab[i].mode = O\_APPEND; // w+  fTab[i].locPoint = 0; // начало файла  ofTabP[i].name = names[i];  ofTabP[i].fd = fd;  }  countP = 7;  printTables();  // Шаг 2  printf("\n\nЗакрытие 2 пользовательских файлов процессом-предком (file3.txt и file4.txt)\n");  countP -= 2;  printTables();  // Шаг 3  printf("\n\nПорождение процесса-потомка через fork()\n");  copyTables();  printTables();  printf("Закрытие 1 пользовательского файла процессом-потомком (file2.txt)\n");  countC--;  fTab[4].linksCount--;  printTables();  // Шаг 4  printf("\n\nЗавершается процесс-потомок. После каждого из этапов печатаются таблицы файлов и открытых файлов для обоих процессов\n");  closeChildProcessFiles();  printTables();  } |

**Содержимое файла Makefile**

|  |
| --- |
| # Makefile for lab #3  Lab3:  gcc lab3.c -std=gnu99 -o lab3  clean:  rm lab3 |

1. Результат работы программы

|  |
| --- |
| Открытие стандартных файлов ввода-ввывода и 4 пользовательских файлов родительским процессом  Таблица файлов:  name links mode locP ref  file1.txt 1 1024 0 6300256  file2.txt 1 1024 0 6300416  file3.txt 1 1024 0 6300576  file4.txt 1 1024 0 6300736  Таблица открытых файлов:  name fd ref  stdin 0 6299776  stdout 1 6299936  stderr 2 6300096  file1.txt 3 6300256  file2.txt 5 6300416  file3.txt 15 6300576  file4.txt 16 6300736  Закрытие 2 пользовательских файлов процессом-предком (file3.txt и file4.txt)  Таблица файлов:  name links mode locP ref  file1.txt 1 1024 0 6300256  file2.txt 1 1024 0 6300416  Таблица открытых файлов:  name fd ref  stdin 0 6299776  stdout 1 6299936  stderr 2 6300096  file1.txt 3 6300256  file2.txt 5 6300416  Порождение процесса-потомка через fork()  Таблица файлов:  name links mode locP ref  file1.txt 2 1024 0 6300256  file2.txt 2 1024 0 6300416  Таблица открытых файлов:  name fd ref  stdin 0 6299776  stdout 1 6299936  stderr 2 6300096  file1.txt 3 6300256  file2.txt 5 6300416  stdin 0 6299776  stdout 1 6299936  stderr 2 6300096  file1.txt 3 6300256  file2.txt 5 6300416  Закрытие 1 пользовательского файла процессом-потомком (file2.txt)  Таблица файлов:  name links mode locP ref  file1.txt 2 1024 0 6300256  file2.txt 1 1024 0 6300416  Таблица открытых файлов:  name fd ref  stdin 0 6299776  stdout 1 6299936  stderr 2 6300096  file1.txt 3 6300256  file2.txt 5 6300416  stdin 0 6299776  stdout 1 6299936  stderr 2 6300096  file1.txt 3 6300256  Завершается процесс-потомок. После каждого из этапов печатаются таблицы файлов и открытых файлов для обоих процессов  Таблица файлов:  name links mode locP ref  file1.txt 1 1024 0 6300256  file2.txt 1 1024 0 6300416  Таблица открытых файлов:  name fd ref  stdin 0 6299776  stdout 1 6299936  stderr 2 6300096  file1.txt 3 6300256  file2.txt 5 6300416 |