Міністерство освіти і науки України Національний університет “Львівська Політехніка”

Лабораторна робота №19

з дисципліни «Програмування частина № 2»

Виконав:

Студент групи АП-11

Іщенко Василь

Прийняв:

Чайковський І.Б.

Львів – 2024

Тема роботи: Дослідження способів організації потокового уведення/виведення в мові програмування С.

Мета роботи: Дослідження способів створення, оновлення та оброблення файлів потокового уведення/виведення даних у мові С.

Попередні відомості

Зберігання даних у змінних і масивах є тимчасовим; всі ці дані втрачаються при завершенні роботи програми. Для постійного зберігання великих об'ємів даних використовуються файли. Комп'ютери зберігають дані на пристроях вторинної пам'яті, головним чином дискових пристроях. Комп'ютер обробляє елементи даних в двійковому вигляді, тобто у вигляді комбінацій нулів і одиниць. Для програмістів обтяжливо працювати з даними низького рівня, якими є біти. Замість цього програмісти вважають за краще працювати з даними у вигляді десяткових цифр, букв, і спеціальних знаків, які називаються символами. Подібно до того, як символи складаються з бітів, поля складаються з символів. Поле є групою символів, які передають значення. Оброблювані комп'ютерами елементи утворюють ієрархію даних, в якій елементи даних стають більші за розміром і складніші за структурою в міру просування від бітів до символів (байтів), полів і так далі. Існує багато способів організації записів у файлі. Найбільш популярний з них називається послідовним файлом, в якому записи, як правило, зберігаються в порядку, що визначається за певним правилом. У файлі нарахування заробітної плати, наприклад, записи зберігалися б впорядкованими за табельним номером. Мова С розглядає будь-який файл як послідовний потік байтів. Кожен файл закінчується маркером кінця файлу, або особливим байтом, визначеним у програмі, що працює з файлом. Коли файл відкривається, йому ставиться у відповідність потік. На початку виконання програми автоматично відкриваються три файли і пов’язані з ними потоки - стандартний ввід, стандартний вивід і стандартна помилка. Потоки забезпечують канали передачі даних між файлами і програмами. Наприклад, стандартний потік уведення дозволяє програмі зчитувати дані з клавіатури, а стандартний потік виведення дозволяє виводити дані на екран.

1. Дослідити та дати пояснення прикладів, викладених нижче.

```
//А)
#include <stdio.h>
#include <windows.h>
int main() {
SetConsoleCP(65001);
SetConsoleOutputCP(65001);
FILE *in; //Опис вказівника на файл
int ch;
if(in=fopen("proba.txt","r")!=NULL){
        while((ch=getc(in)!=EOF))// Отримується символ із in
        printf("%c",ch);
        putc(ch,stdout);// Виведення символа в стандартний потік на екран.
        fclose(in); }
else
printf("Файл proba не відкривається \n"); }
//Б)
#include <stdio.h>
int main( ) {
FILE *ff;
int base;
ff = fopen("proba.txt" , "r");
// відкривається файл із іменем sam, який ідентифікується зі вказівником на ff .
fscanf( ff, " %d" , &base);   // ff вказує на файл із іменем sam
fclose(ff);
ff = fopen("data", "a");   // доповнення
fprintf( ff, "sam is %d.\n", base);   /* ff вказує на data */
fclose(ff); }

//В)
#include <stdio.h>
#define LINE 80
int main( ){
FILE *ff;
char *string[LINE];
ff = fopen(" opus", "r");
while ( fgets(string, LINE, ff) != NULL)
 puts(string); }

//Г)
Приклад читання форматованих даних з файлу "C:\\temp\\sample.txt":
/* Читання форматованих даних за допомогою функції fscanf(). */
#include <stdlib.h>
#include <stdio.h>
int main(){
 int f1, f2, f3, f4, f5;
 FILE *fp;
fp = fopen("C:\\temp\\sample.txt", "r"); /*Відкриття файлу в режимі
читання*/
/*Читання з файлу */
fscanf(fp, "%d\n%d\n%d\n%d\n%d\n", &f1, &f2, &f3, &f4, &f5);
printf("The values are %d, %d, %d, %d, %d \n.",f1, f2, f3, f4, f5);
fclose(fp); /* Закриття файлу fp*/}
```

Завдання 3

```
//Запис даних в файл proba.txt
#include <stdio.h>
int main (void)
{
 FILE *pf;
 int k;
if ((pf = fopen ("proba.txt","w")) ==NULL){
 perror("proba.txt");
 return 1;
 }
 for (k=0; k<=5; k++)
 fprintf(pf, "%d %d\n", k, k*k*k*k);
 fclose(pf);
 return 0;
 }
//Читання даних із файлу proba.txt
#include <stdio.h>
int main (void)
{
FILE *pf;
int n, nn, l;
if ((pf = fopen ("proba.txt","r")) ==NULL)
{
   perror("proba.txt");
   return 1;
}
for (l=0; l<=5; l++){
fscanf(pf, "%d %d\n", &n, &nn);}
fclose(pf);
return 0;
}
```

Завдання 4

```
//Запис даних в файл proba.txt
#include <stdio.h>
int main (void)
{
int a[8]={1,2,3,4,5,6,7,8};
FILE *pf;
int k;
if ((pf = fopen ("proba.txt","w")) ==NULL)
{
perror("proba.txt");
return 1;
}
for (k=0; k<8; k++)
fprintf(pf, "%d\n", a[k]);
fclose(pf);
return 0;
}
```

```
//Читання даних із файлу proba.txt
#include <stdio.h>
int main (void)
{
int b[8]={};
FILE *pf;
int n, l, i;
if ((pf = fopen ("proba.txt","r")) ==NULL)
{
  perror("proba.txt");
  return 1;
}
for (l=0; l<8; l++){
fscanf(pf, "%d\n", &b[l]);}
fclose(pf);
for (i = 0; i<8; i++){
printf("%d\n", b[i]);}
return 0;
}
```

Висновок: Дослідження способів створення, оновлення та оброблення файлів потокового уведення/виведення даних у мові С.