Московский Авиационный Институт

(Национальный Исследовательский Университет)

Кафедра: 806 «Вычислительная математика и программирование»

Факультет: «Прикладная математика и физика»

Дисциплина: «Операционные системы»

Курсовой проект.

Тема:

«Poker.pipes»

Группа: 8О-204Б

Студент: Лебедев Т.А.

Вариант: №

Оценка:

Подпись:

Москва

2016

Source.cpp

typedef struct arr{

int b[6];

};

int combination(int\* a);

arr find\_m(int \*a);

int combination(int\* a)

{

arr comb;

int z = 0;

int i = 0;

bool m\_street, house, dole, b\_street,n;

b\_street = m\_street = house = dole = n = false;

comb = find\_m(a);

printf("\n");

while (i != 6)

{

if (comb.b[i] == 0)

{

i++;

}

if (comb.b[i] == 5)

{

return 9;

}

if (comb.b[i] == 4)

{

return 8;

}

if (comb.b[i] == 3)

{

z = i;

while (z != 6)

{

z++;

if (comb.b[z] == 2)

{

house = true;

}

}

if (house == true)

{

return 7;

}

else

{

return 4;

}

}

if (comb.b[i] == 2)

{

z = i;

while (z != 6)

{

z++;

if (comb.b[z] == 3)

{

house = true;

}

if (comb.b[z] == 2)

{

dole = true;

}

}

if (house == true)

{

return 7;

}

if (dole == true)

{

return 3;

}

else { return 2; }

}

if (comb.b[0] == 1 && comb.b[1] == 1 && comb.b[2] == 1 && comb.b[3] == 1 && comb.b[4] == 1)

{

return 5;

}

if (comb.b[1] == 1 && comb.b[2] == 1 && comb.b[3] == 1 && comb.b[4] == 1 && comb.b[5] == 1)

{

return 6;

}

i++;

}

return 1;

}

arr find\_m(int \*a)

{

arr s;

s.b[0] = s.b[1] = s.b[2] = s.b[3] = s.b[4] = s.b[5] = 0;

for (int i = 0; i<6; i++)

{

switch (a[i])

{

case 1:{

s.b[0]++; break; }

case 2:{

s.b[1]++; break; }

case 3:{

s.b[2]++; break; }

case 4:{

s.b[3]++; break; }

case 5:{

s.b[4]++; break; }

case 6:{

s.b[5]++; break; }

}

}

return s;

}

void res(int a){

switch (a)

{

case 1:

{printf("Nothing\n"); break; }

case 2:

{printf("Pair\n"); break; }

case 4:

{printf("Set\n"); break; }

case 8:

{printf("Karre\n"); break; }

case 9:

{printf("Poker\n"); break; }

case 5:

{printf("Small Street\n"); break; }

case 6:

{printf("Big Street\n"); break; }

case 7:

{printf("Full-house\n"); break; }

case 3:

{printf("Double pair\n"); break; }

}

}

void display (int a)

{

switch (a)

{

case 1:

{printf("\_\_\_\n"); printf("| . |\n"); printf("| |\n");printf("\_\_\_\t");break; }

case 2:

{printf("\_\_\_\n"); printf("|. .|\n"); printf("| |\n");printf("\_\_\_\t");break; }

case 3:

{printf("\_\_\_\n"); printf("|...|\n"); printf("| |\n");printf("\_\_\_\t");break; }

case 4:

{printf("\_\_\_\n"); printf("|...|\n"); printf("| . |\n");printf("\_\_\_\t");break; }

case 5:

{printf("\_\_\_\n"); printf("|...|\n"); printf("|. .|\n");printf("\_\_\_\t");break; }

case 6:

{printf("\_\_\_\n"); printf("|...|\n"); printf("|...|\n");printf("\_\_\_\t");break; }

}

}

SonPipe (1).cpp

#include <time.h>

#include <windows.h>

#include <stdio.h>

#include <math.h>

#include <tchar.h>

#include <conio.h>

#include <string.h>

#include <stdlib.h>

#include "Source.cpp"

typedef struct poker{

int dices[5];

};

int \_tmain(int argc, \_TCHAR\* argv[])

{

srand(time(NULL));

int i;

int number;

int z;

poker pl1;

int dice;

HANDLE Son;

LPCSTR AdresNameFile = "\\\\.\\pipe\\MyPipe";

Sleep (12);

Son = CreateFile(AdresNameFile, GENERIC\_READ | GENERIC\_WRITE , 0, NULL, OPEN\_EXISTING,FILE\_ATTRIBUTE\_NORMAL, NULL);

if (Son == INVALID\_HANDLE\_VALUE)

{

printf("I'm Son.\n Daddy doesn't createeeeeeeeee");

getchar();

printf("NNNNNNOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO");

return GetLastError();

};

// bool connect = ConnectNamedPipe(Son, &l);

/\*if (connect == false)

{

int i = GetLastError();

if (i == 997 || i == 536)

{

}

else {

return GetLastError();

};

};\*/

printf("Here we go!\n");

for(i=0;i<5;i++){

pl1.dices[i]=rand() %6 +1;

display(pl1.dices[i]);}

res (combination(pl1.dices));

printf("Do you want change some of them? 1/0 ");

int ch;

scanf("%d", &ch);

// while (ch!=1 ||ch!=0){

// scanf("%d", &ch);}

if(ch==1)

{printf("Which ones don't suit you? Enter number of items for change\n");

scanf("%d", &number);

for(int i=0;i<number;i++)

{printf("Enter number: ");

scanf("%d", &dice);

dice--;

pl1.dices[dice]=rand() %6 +1;

}

for(i=0;i<5;i++){

display(pl1.dices[i]);}

res (combination(pl1.dices));

}

int a=(combination(pl1.dices));

DWORD in;

if (!ReadFile(Son, &z,sizeof(int), &in , NULL))

/\*{

printf("I CAN'T READ FILE");

return GetLastError();

};\*/

/\*Son = CreateFile(AdresNameFile, GENERIC\_READ | GENERIC\_WRITE, FILE\_SHARE\_READ | FILE\_SHARE\_WRITE, NULL, TRUNCATE\_EXISTING, FILE\_ATTRIBUTE\_NORMAL, NULL);

if (Son == INVALID\_HANDLE\_VALUE)

{

printf("Son doesn't open file");

getchar();

printf("NNNNNNOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO");

ExitProcess(1);

return 1;

};\*/

printf("%d\n", &z);

if(z>a)

{printf("YOU LOOSE!\n");}

if(z<a)

{printf("YOU WIN!\n");}

if(a==z)

{printf("IT'S A DRAW!\n");}

DWORD wr;

WriteFile(Son, &a,sizeof(int), &wr , NULL);

Sleep(100);

getch();

ExitProcess(1);

return 0;

}

LPCVOID Error(int i)

{

LPCVOID Errors;

if (i == 1)

{

Errors = TEXT("you bad write \n");

return Errors;

};

return 0;

};

fatherpipe (1).cpp

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <time.h>

#include <windows.h>

#include <stdio.h>

#include <math.h>

#include <tchar.h>

#include <conio.h>

#include <string.h>

#include <stdlib.h>

#include "Source.cpp"

typedef struct poker{

int dices[5];

};

int \_tmain(int argc, \_TCHAR\* argv[])

{

srand(time(NULL));

int i;

int number;

poker pl1;

int dice;

printf("Waiting for next player\n");

HANDLE Daddy;

LPCSTR AdresNameFile = ("\\\\.\\pipe\\MyPipe");

//LPWSTR AdresNameFile =TEXT ("C:\\Users\\Arkadii\\Desktop\\OS\\Data.txt");

Daddy = CreateNamedPipe(AdresNameFile, PIPE\_ACCESS\_DUPLEX,

PIPE\_TYPE\_MESSAGE | PIPE\_READMODE\_MESSAGE | PIPE\_WAIT,

1, 128, 128,

PIPE\_UNLIMITED\_INSTANCES,

NULL);

if (Daddy == INVALID\_HANDLE\_VALUE)

{

printf("Daddy's PIPE doesn't createeeeeeeeee");

printf(" NNNNNNOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO");

system("pause");

return GetLastError();

}

PROCESS\_INFORMATION ProcessInfo; //This is what we get as an [out] parameter

ZeroMemory(&ProcessInfo, sizeof(PROCESS\_INFORMATION));

STARTUPINFO StartupInfo; //This is an [in] parameter

ZeroMemory(&StartupInfo, sizeof(StartupInfo));

StartupInfo.cb = sizeof(STARTUPINFO); //Only compulsory field

bool process = CreateProcess("C:\\Users\\L2(client).exe",

NULL,

NULL, NULL, true,

CREATE\_NEW\_CONSOLE,// CREATE\_NEW\_CONSOLE|CREATE\_SUSPENDED

NULL, NULL,

&StartupInfo,

&ProcessInfo);

WaitForSingleObject(ProcessInfo.hProcess, 5);

bool connect = ConnectNamedPipe(Daddy, NULL);

if(connect == true)

{printf("We just find someone!\n");}

if (connect == false)

{

int i = GetLastError();

if (i == 997 || i == 536)

{

}

else {

return GetLastError();

};

};

printf("Find your destiny!\n");

printf("Here we go!\n");

for(i=0;i<5;i++){

pl1.dices[i]=rand() %6 +1;

display(pl1.dices[i]);}

res (combination(pl1.dices));

printf("Do you want change some of them? 1/0 ");

int ch;

scanf("%d", &ch);

// while (ch!=1 ||ch!=0){

// scanf("%d", &ch);}

if(ch==1)

{printf("Which ones don't suit you? Enter number of items for change\n");

scanf("%d", &number);

for(int i=0;i<number;i++)

{printf("Enter number: ");

scanf("%d", &dice);

dice--;

pl1.dices[dice]=rand() %6 +1;

}

for(i=0;i<5;i++){

display(pl1.dices[i]);}

res (combination(pl1.dices));

}

DWORD wr;

int a=(combination(pl1.dices));

WriteFile(Daddy, &a,sizeof(int), &wr , NULL);

Sleep(100);

DWORD r; int z;

ReadFile(Daddy, &z,sizeof(int), &r , NULL);

if(z>(combination(pl1.dices)))

{printf("YOU LOOSE!\n");}

if(z<(combination(pl1.dices)))

{printf("YOU WIN!\n");}

if(z==(combination(pl1.dices)))

{printf("IT'S A DRAW!\n");}

getch();

/\* DWORD written;

if (!WriteFile(Daddy, &z, sizeof(node), &written, NULL))

{

printf("Error! Can not write in file\n");

system("pause");

return GetLastError();

}

Sleep(1000);

ReadFile(Daddy, &z,sizeof(node), &written , NULL);

CloseHandle(Daddy);

system("pause");\*/

return 0;

}





