Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное   
образовательное учреждение высшего образования   
«Самарский национальный исследовательский университет   
имени академика С.П. Королева»

Институт информатики и кибернетики

Кафедра технической кибернетики

Отчет по лабораторной работе №3

Дисциплина: «Операционные системы»

Тема «Межпроцессное взаимодействие в операционной системе Ubuntu»

Выполнил: Белов А. А.

Группа: 6205-010302D

Самара, 2023

**Задание на лабораторную работу**

Перевод из произвольной (выбирает пользователь) системы счисления в 10-чную и обратно.

**Текст программы**

// вариант 5

#include <iostream>

#include <unistd.h>

#include <sys/wait.h>

#include <cstdlib>

#include <cstring>

#include <string>

#include <cmath>

using namespace std;

int toDecimal(string num, int ss){

int result = 0;

int count = 1;

for (int i = num.size() -1; i >= 0; --i){

if (isdigit(num[i]))

result += (num[i] - '0') \* count;

else

result += (num[i] - 'A' + 10) \* count;

count \*= ss;

}

return result;

}

string fromDecimal(int num, int ss){

string result = "";

while(num){

int ost = num % ss;

if (ost < 10)

result = (char)('0' + ost) + result;

else

result = (char)('A' + (ost - 10)) + result;

num /= ss;

}

return result;

}

string cheak\_ss(){ // проверяет наличие цифр в string (или end и --help)

string ss;

int n;

while(true){

cout << "Select number system (or <end>, to exit the program): ";

cin >> ss;

if (ss == "--help" || ss == "end")

return ss;

try {

n = stoi(ss);

}

catch(const invalid\_argument e) {

cout << "Error int input! " << endl;

cin.clear();

}

if(n > 1 && n < 36)

return ss;

}

}

string cheak\_num(int s){

int ss = s;

string num;

bool cur = true;

while(cur){

cout << "Enter the number: ";

cin >> num;

cur = false;

try {

if (stoi(num) <0)

cout << "Error num < 0" << endl;

}

catch(exception){continue;}

for (int i = 0; i < num.length();++i){

char c = num[i];

/\*if ( stoi(num) <= 0){

cout << "Does not the ss!" << endl;

cur = true;

}\*/

if (isalnum(c)) {// проверяем является ли символ цифрой или буквой

int digit;

if (isdigit(c))

digit = c - '0';

else

digit = toupper(c) - 'A' + 10;

if (digit >= ss ){

cout << "Does not the ss!" << endl;

cur = true;

break;

}

}

else{

cur = true;

}

}

}

return num;

}

int main(){

int clientToServer[2]; // канал для передачи данных и команд от клиента к серверу

int serverToClient[2]; // канал для передачи результата обратно от сервера к клиенту

// 0 - read

// 1 - write

if(pipe(clientToServer) == -1 || pipe(serverToClient) == -1){ // создаём и проверяем канал на корректное созданеи

cout << "Error when creating a chanel!" << endl;

return 1;

}

pid\_t pid = fork(); // создание процесса-потомка

if (pid == -1){

cout << "Error when creating the process!" << endl;

return 1;

}

else{

if (pid == 0){ // серверный процесс (дочерний/потомок)

close(clientToServer[1]); // закрытие канала для записи

close(serverToClient[0]); // закрытие канала для чтения

while(true){

sleep(1);

char ssbuf[1024];

read(clientToServer[0], ssbuf, sizeof(ssbuf)); // считываем систему счисления

if (strcmp(ssbuf, "--help") == 0){

const char\* help = "help";

write(serverToClient[1], help, strlen(help)+1);

break;

}

if (strcmp(ssbuf, "end") == 0)

break;

int ss = stoi(ssbuf);

//cout << "ss: " << ss << endl;

char numbuf[1024];

read(clientToServer[0], numbuf, sizeof(numbuf)); // считываем число

//cout << numbuf << endl;

string num = numbuf;

int to\_10 = toDecimal(num, ss); // в десятичную систему счисления

//cout << "10: "<< res << endl;

string back\_ss = fromDecimal(to\_10, ss); // обратно

//cout << ss << ": " << res1 << endl;

write(serverToClient[1], &to\_10, sizeof(int)); // 10-сс

const char\* back\_ss\_buf = back\_ss.c\_str();

write(serverToClient[1], back\_ss\_buf, strlen(back\_ss\_buf)+1); // n-сс

}

close(clientToServer[0]);

close(serverToClient[1]);

\_exit(0); // немедленное завершение процесса (не вызывает деструкторы)

}

else{ // клиентский процесс

close(clientToServer[0]); // закрытие канала чтения

close(serverToClient[1]); // закрытие канала записи

while(true){ // end выходить из цикла

string ss;

string num;

ss = cheak\_ss();//ввод и контроль ввода для сс

const char\* ssbuf = ss.c\_str();

if (write(clientToServer[1], ssbuf, strlen(ssbuf) + 1) == -1) // отправляем на сервер систему счисления

cout << "Error while sending to server! " << endl;

if (ss == "--help"){

sleep(1);

char key[1024];

read(serverToClient[0], key, sizeof(key));

cout << key <<endl;

break;

}

if (ss == "end")

break;

num = cheak\_num(stoi(ss));

//cout << "Enter the number: ";

//cin >> num;

const char\* numbuf = num.c\_str();

sleep(1);

write(clientToServer[1], numbuf, strlen(numbuf) + 1); // отправляем на сервер число

int to\_10;

read(serverToClient[0], &to\_10, sizeof(int));

char back\_ss[1024];

read(serverToClient[0], back\_ss, sizeof(back\_ss));

cout << "Result to 10-ss: " << to\_10 << endl;

cout << "Result back: " << back\_ss << endl;

cout << endl << endl;

}

close(clientToServer[1]);

close(serverToClient[0]);

wait(nullptr); // ждёт завершения дочернего процесса

exit(0); // завершает процесс (предварительно вызывает деструкторы)

}

}

return 0;

}