**Код программы**

#include<cstdlib>

#include<iostream>

#include<unistd.h>

#include<ctime>

#include<string.h>

#include<sys/wait.h>

#include<sys/errno.h>

using namespace std;

int firstProcces();

int secondProcces();

int thirdProcces();

int fourthProcces();

int execute(int);

string tab(unsigned short);

int main(int argc, char \*\*argv){

if(argc != 5){

cout << "ERROR: Program need 4 parameters!\n";

return EXIT\_FAILURE;

}

cout << "PARENT BEGIN\n";

int pid(-1);

int status;

for(int i = 1; i <= 4; ++i){

pid = fork();

if(pid == 0){

return execute(atoi(argv[i]));

}else{

cout << "Process created\n";

int ch\_pid = wait(&status);

cout << "Process with PID " << ch\_pid << " is over with code " << status << "\n\n";

}

}

cout << "THE END\n";

return EXIT\_SUCCESS;

}

//----------------------------

string tab(unsigned short n){

if(n != 0){

string str(" ");

str += tab(n-1);

return str;

}

return "";

}

//----------------------------

int firstProcces(){

time\_t current\_time = time(NULL);

cout << tab(1) << "I'm first (1), my PID: " << getpid() << " my PPID: " << getppid() << ". Current time is: " << ctime(&current\_time);

cout << tab(1) << "I'm calling for programm named \"first\" (custom)\n";

cout << tab(1) << "Executed programm output:\n";

char \* const args[5] = {"first", "arg1", "arg2", "arg3", NULL};

int status;

int pid = fork();

if(pid == 0){

execv("first", args);

char\* buf = new char[100];

cout << tab(2) << "FirstProcess::execv::ERROR::" << strerror\_r(errno, buf, 100) << "\n";

return EXIT\_FAILURE;

}else{

if(wait(&status) == -1){

char\* buf = new char[100];

cout << tab(2) << "FirstProcess::wait::ERROR::" << strerror\_r(errno, buf, 100) << "\n";

return EXIT\_FAILURE;

}

if(status != 0){

return EXIT\_FAILURE;

}

current\_time = time(NULL);

cout << tab(1) << "First(1) finished! Current time is: " << ctime(&current\_time);

}

return EXIT\_SUCCESS;

}

int secondProcces(){

time\_t current\_time = time(NULL);

cout << tab(1) << "I'm second (2), my PID: " << getpid() << " my PPID: " << getppid() << ". Current time is: " << ctime(&current\_time);

cout << tab(1) << "I'm calling for programm named \"second\" (custom)\n";

cout << tab(1) << "Executed programm output:\n";

int status;

int pid = fork();

if(pid == 0){

execl("second", "second", "-help", NULL);

char\* buf = new char[100];

cout << tab(2) << "SecondProcess::execl::ERROR::" << strerror\_r(errno, buf, 100) << "\n";

//perror("execv");

return EXIT\_FAILURE;

}else{

if(wait(&status) == -1){

char\* buf = new char[100];

cout << tab(2) << "SecondProcess::wait::ERROR::" << strerror\_r(errno, buf, 100) << "\n";

return EXIT\_FAILURE;

}

if(status != 0){

return EXIT\_FAILURE;

}

current\_time = time(NULL);

cout << tab(1) << "Second (1) finished! Current time is: " << ctime(&current\_time);

}

return EXIT\_SUCCESS;

}

int thirdProcces(){

time\_t current\_time = time(NULL);

cout << tab(1) << "I'm third (3), my PID: " << getpid() << " my PPID: " << getppid() << ". Current time is: " << ctime(&current\_time);

cout << tab(1) << "I'm calling for programm named \"vi\"\n";

cout << tab(1) << "Executed programm output:\n";

char \* const args[4] = {"vi", "-o", "openme.txt", NULL};

int status;

int pid = fork();

if(pid == 0){

execvp("vi", args);

char\* buf = new char[100];

cout << tab(2) << "ThirdProcess::execvp::ERROR::" << strerror\_r(errno, buf, 100) << "\n";

return EXIT\_FAILURE;

}else{

if(wait(&status) == -1){

char\* buf = new char[100];

cout << tab(2) << "ThirdProcess::wait::ERROR::" << strerror\_r(errno, buf, 100) << "\n";

return EXIT\_FAILURE;

}

if(status != 0){

return EXIT\_FAILURE;

}

current\_time = time(NULL);

cout << tab(1) << "Third (1) finished! Current time is: " << ctime(&current\_time);

}

return EXIT\_SUCCESS;

}

int fourthProcces(){

time\_t current\_time = time(NULL);

cout << tab(1) << "I'm fourth (4), my PID: " << getpid() << " my PPID: " << getppid() << ". Current time is: " << ctime(&current\_time);

cout << tab(1) << "I'm calling for programm named \"ls\"\n";

cout << tab(1) << "Executed programm output:\n";

int status;

int pid = fork();

if(pid == 0){

execlp("ls", "ls", "-l", NULL);

char\* buf = new char[100];

cout << tab(2) << "FourthProcess::execvp::ERROR::" << strerror\_r(errno, buf, 100) << "\n";

return EXIT\_FAILURE;

}else{

if(wait(&status) == -1){

char\* buf = new char[100];

cout << tab(2) << "FourthProcess::wait::ERROR::" << strerror\_r(errno, buf, 100) << "\n";

return EXIT\_FAILURE;

}

if(status != 0){

return EXIT\_FAILURE;

}

current\_time = time(NULL);

cout << tab(1) << "Fourth (4) finished! Current time is: " << ctime(&current\_time);

}

return EXIT\_SUCCESS;

}

int execute(int n){

int ret\_code;

switch (n)

{

case 1:

ret\_code = firstProcces();

break;

case 2:

ret\_code = secondProcces();

break;

case 3:

ret\_code = thirdProcces();

break;

case 4:

ret\_code = fourthProcces();

break;

default:

break;

}

return ret\_code;

}

**Пример работы программы**

****