Учреждения образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

[Кафедра программной инженерии](https://www.belstu.by/fakultety/fit/vm)

Специальность 1-40 01 01 Программное обеспечение информационных технологий

**Отчёт по лабораторной работе №7**

Дисциплина: Операционные системы

Тема: Компьютерное время

Выполнила:

студентка 3 курса 5 группы

Городилина Анастасия Сергеевна

Минск 2024

**Задание 01**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <chrono>

#include <iomanip>

#include <ctime>

using namespace std;

int main()

{

auto now = chrono::system\_clock::now();

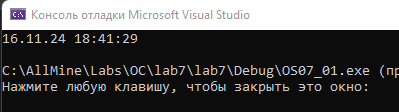
time\_t now\_c = chrono::system\_clock::to\_time\_t(now);

tm\* local\_time = localtime(&now\_c);

cout << put\_time(local\_time, "%d.%m.%y %H:%M:%S") << endl;

return 0;

}



**Задание 02**

#include <iostream>

#include <chrono>

#include <ctime>

#include <thread>

using namespace std;

atomic<int> i = 0;

void printIterationsAfterTimeFive() {

this\_thread::sleep\_for(chrono::seconds(5));

cout << "Прошло 5 секунд: " << i << endl;

}

void printIterationsAfterTimeTen() {

this\_thread::sleep\_for(chrono::seconds(10));

cout << "Прошло 10 секунд: " << i << endl;

}

void exitProcess() {

this\_thread::sleep\_for(chrono::seconds(15));

cout << "Прошло 15 секунд, программа завершает работу: " << i << endl;

exit(0);

}

int main()

{

setlocale(LC\_ALL, "rus");

thread t1(printIterationsAfterTimeFive);

thread t2(printIterationsAfterTimeTen);

thread t3(exitProcess);

t1.detach();

t2.detach();

t3.detach();

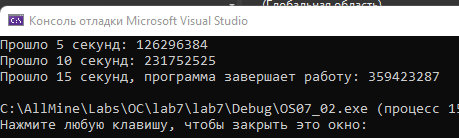
while (true) {

i++;

}

return 0;

}



**Задание 03**

#include <iostream>

#include <chrono>

#include <ctime>

#include <thread>

#include <Windows.h>

using namespace std;

int i = 0;

bool flag = true;

void printIterationsAfterTime();

void timerThread(HANDLE htimer);

void exitProcess();

int main()

{

setlocale(LC\_ALL, "rus");

HANDLE htimer = CreateWaitableTimer(NULL, FALSE, NULL);

LARGE\_INTEGER it;

it.QuadPart = -3000 \* 10000;

if (!SetWaitableTimer(htimer, &it, 3000, NULL, NULL, TRUE)) {

cout << "Ошибка создания таймера" << endl;

return 1;

}

thread t(timerThread, htimer);

thread t1(exitProcess);

t.detach();

t1.detach();

while (true) {

i++;

Sleep(100);

}

CancelWaitableTimer(htimer);

CloseHandle(htimer);

return 0;

}

void printIterationsAfterTime() {

cout << "Прошло 3 секунды: " << i << endl;

}

void timerThread(HANDLE htimer) {

while (true) {

WaitForSingleObject(htimer, INFINITE);

if (flag) {

flag = false;

continue;

}

printIterationsAfterTime();

}

}

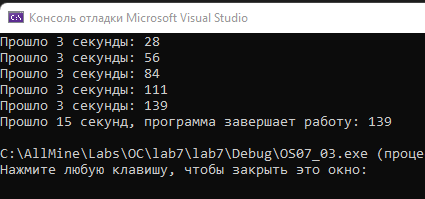
void exitProcess() {

this\_thread::sleep\_for(chrono::seconds(15));

cout << "Прошло 15 секунд, программа завершает работу: " << i << endl;

exit(0);

}



**Задание 04**

OS07\_04x

#include <iostream>

#include <Windows.h>

#include <ctime>

using namespace std;

void findPrimes(HANDLE htimer) {

int num = 2, i, count = 1;

bool isPrime;

time\_t startTime = time(nullptr);

for (num; ; num++) {

isPrime = true;

for (i = 2; i \* i <= num; ++i) {

if (num % i == 0) {

isPrime = false;

break;

}

}

if (isPrime) {

cout << count << ") " << num << endl;

count++;

Sleep(200);

}

if (WaitForSingleObject(htimer, 0) == WAIT\_OBJECT\_0) {

time\_t endTime = time(nullptr);

cout << endTime - startTime << " sec" << endl;

cout << "Завершение программы" << endl;

break;

}

}

}

int main(int argc, char\* argv[])

{

setlocale(LC\_ALL, "rus");

if (argc != 2) {

cout << "Нужно ввести 2 аргумента" << endl;

return 1;

}

int duration = atoi(argv[1]);

HANDLE htimer = CreateWaitableTimer(NULL, FALSE, NULL);

LARGE\_INTEGER li;

li.QuadPart = -duration \* 60 \* 10000000;

if (!SetWaitableTimer(htimer, &li, 0, NULL, NULL, 0)) {

cerr << "Ошибка создания таймера" << endl;

return 1;

}

findPrimes(htimer);

system("pause");

CancelWaitableTimer(htimer);

CloseHandle(htimer);

return 0;

}

OS07\_04

#include <iostream>

#include <Windows.h>

#include <process.h>

#include <vector>

#include <string>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

const wchar\_t\* path=L"C:\\AllMine\\Labs\\ОС\\lab7\\lab7\\Debug\\OS07\_04x.exe";

vector<PROCESS\_INFORMATION> pInfo(2);

vector<STARTUPINFO> sInfo(2);

for (int i = 0; i < 2; ++i) {

ZeroMemory(&sInfo[i], sizeof(STARTUPINFO));

sInfo[i].cb = sizeof(STARTUPINFO);

ZeroMemory(&pInfo[i], sizeof(PROCESS\_INFORMATION));

wchar\_t args[100];

swprintf(args, sizeof(args) / sizeof(wchar\_t), L"%d", i + 1);

wchar\_t command[200];

swprintf(command, sizeof(command) / sizeof(wchar\_t), L"%s %s", path, args);

if (!CreateProcess(NULL, command, NULL, NULL, FALSE, CREATE\_NEW\_CONSOLE, NULL, NULL, &sInfo[i], &pInfo[i])) {

wcout << L"Ошибка при создании процесса: " << GetLastError() << endl;

return 1;

}

}

WaitForSingleObject(pInfo[0].hProcess, INFINITE);

WaitForSingleObject(pInfo[1].hProcess, INFINITE);

for (int i = 0; i < 2; ++i) {

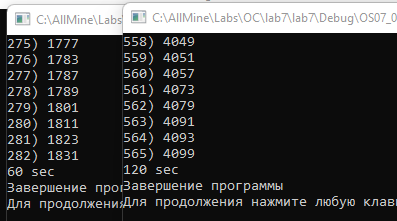
CloseHandle(pInfo[i].hProcess);

CloseHandle(pInfo[i].hThread);

}

return 0;

}



**Задание 05**

#include <stdio.h>

#include <time.h>

int main() {

time\_t now = time(NULL);

struct tm \*local\_time = localtime(&now);

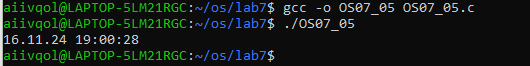
char buffer[80];

strftime(buffer, sizeof(buffer), "%d.%m.%y %H:%M:%S", local\_time);

printf("%s\n", buffer);

return 0;

}



**Задание 06**

#include <stdio.h>

#include <time.h>

#include <unistd.h>

int main() {

long iters = 0;

time\_t startTime, currentTime;

time(&startTime);

while (1) {

iters++;

currentTime = time(NULL);

double elapsedTime = difftime(currentTime, startTime);

if (elapsedTime >= 2.0) {

printf("2 seconds have passed: %ld\n", iters);

printf("Time: %.0f sec\n", elapsedTime);

startTime = currentTime;

}

}

return 0;

}

