Учреждение образования

«Белорусский государственный технологический университет»

**Лабораторная работа №7**

Выполнил:

Студент 3 курса 4 группы ФИТ

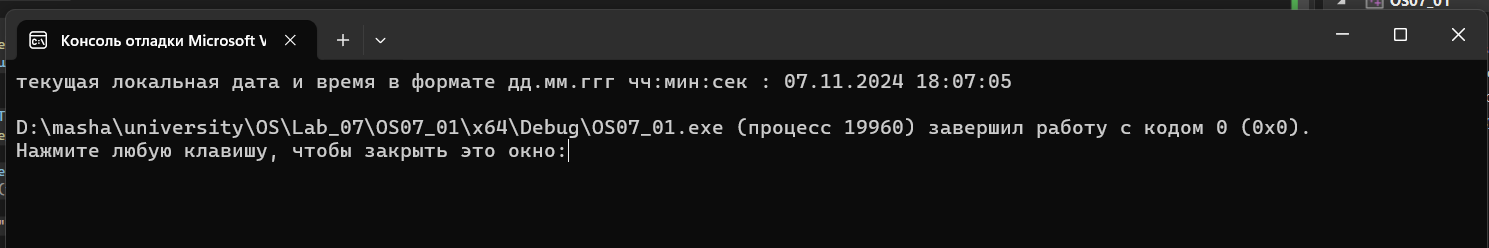
Сосновец Мария Игоревна

2024 г.

**Задание 01. Windows**

1. Разработайте приложение **OS07\_01**.
2. Приложение **OS07\_01** выводит на консоль текущую локальную дату и время в формате ***дд.мм.ггг чч:мин:сек***.

|  |
| --- |
| 1. #include <iostream> 2. #include <ctime> 3. using namespace std; 4. int main() { 5. setlocale(LC\_ALL, "rus"); 6. time\_t currentTime = time(nullptr); 7. tm localTimeStruct; 8. localtime\_s(&localTimeStruct, &currentTime); 9. char timeBuffer[20]; 10. strftime(timeBuffer, sizeof(timeBuffer), "%d.%m.%Y %H:%M:%S", &localTimeStruct); 11. cout << "текущая локальная дата и время в формате дд.мм.ггг чч:мин:сек : " << timeBuffer << endl; 12. return 0; 13. } |



**Задание 02. Windows**

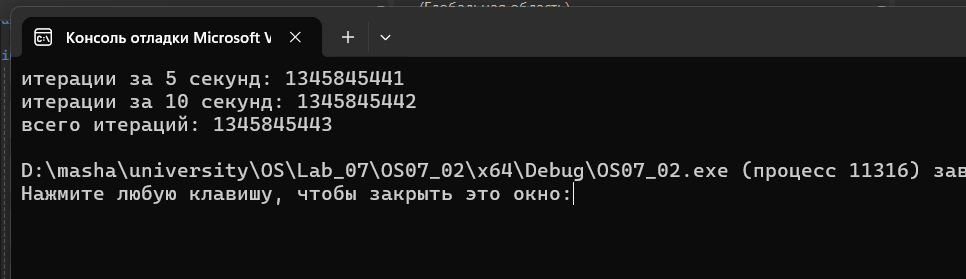
Разработайте приложение **OS07\_02,** выполняющее бесконечный цикл.

В теле цикла подсчитывается количество итераций.

Выведите на консоль значения счетчика итераций через 5 сек. и 10 сек.

Корректно завершите работу цикла и приложения через 15 сек., выведите итоговое значение счетчика итераций.

|  |
| --- |
| #include <iostream>  #include <windows.h>  using namespace std;  int main() {  setlocale(LC\_ALL, "rus");  int iterationCount = 0;  DWORD startTime = GetTickCount();  while (true) {  iterationCount++;  DWORD currentTime = GetTickCount();  DWORD elapsedTime = currentTime - startTime;  if (elapsedTime >= 5000 && elapsedTime < 10000) {  cout << "итерации за 5 секунд: " << iterationCount << endl;  Sleep(5000);  }  if (elapsedTime >= 10000 && elapsedTime < 15000) {  cout << "итерации за 10 секунд: " << iterationCount << endl;  Sleep(5000);  }  if (elapsedTime >= 15000) {  cout << "всего итераций: " << iterationCount << endl;  break;  }  }  return 0;  } |



**Задание 03. Windows**

**Указание: самостоятельно освойте и примените периодический ожидающий таймер**

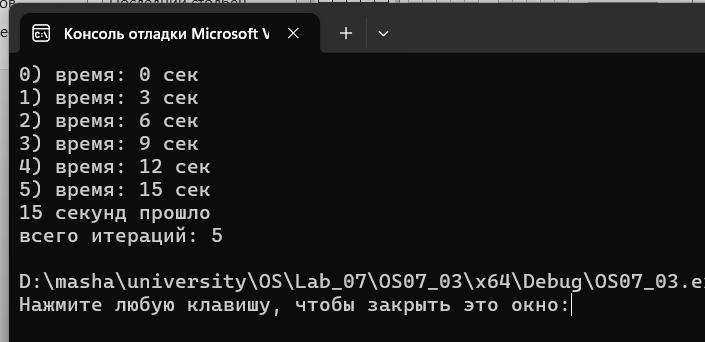
Разработайте приложение **OS07\_03,** выполняющее бесконечный цикл.

В теле цикла с задержкой подсчитывается количество итераций.

Выведите на консоль значения счетчика итераций каждые 3 сек.

Корректно завершите работу цикла и приложения через 15 сек., выведите итоговое значение счетчика итераций.

|  |
| --- |
| #include <iostream>  #include <windows.h>  #define SECOND 10000000  using namespace std;  int main()  {  setlocale(LC\_ALL, "rus");  HANDLE htimer = CreateWaitableTimer(NULL, FALSE, NULL);  if (htimer == NULL) {  cerr << "таймер не создался" << endl;  return 1;  }  LARGE\_INTEGER li;  li.QuadPart = -3 \* SECOND;  DWORD start = GetTickCount();  int iterationCount = 0;  while (true) {  if (!SetWaitableTimer(htimer, &li, 0, NULL, NULL, FALSE)) {  cerr << "ошибка уставноелния таймера" << endl;  CloseHandle(htimer);  return 1;  }  DWORD elapsed = (GetTickCount() - start) / 1000;  cout << iterationCount << ") время: " << elapsed << " сек" << endl;  if (elapsed >= 15) {  cout << "15 секунд прошло" << endl;  cout << "всего итераций: " << iterationCount << endl;  break;  }  WaitForSingleObject(htimer, INFINITE);  iterationCount++;  }  CloseHandle(htimer);  return 0;  } |



**Задание 04. Windows**

Разработайте приложение **OS07\_04,** запускающее два одинаковых дочерних процесса **OS07\_04\_X**.

Процессы **OS07\_04\_X** вычисляют и выводят на консоль (каждый в свою) пронумерованный ряд простых положительных чисел (простое число делится нацело только на себя и 1).

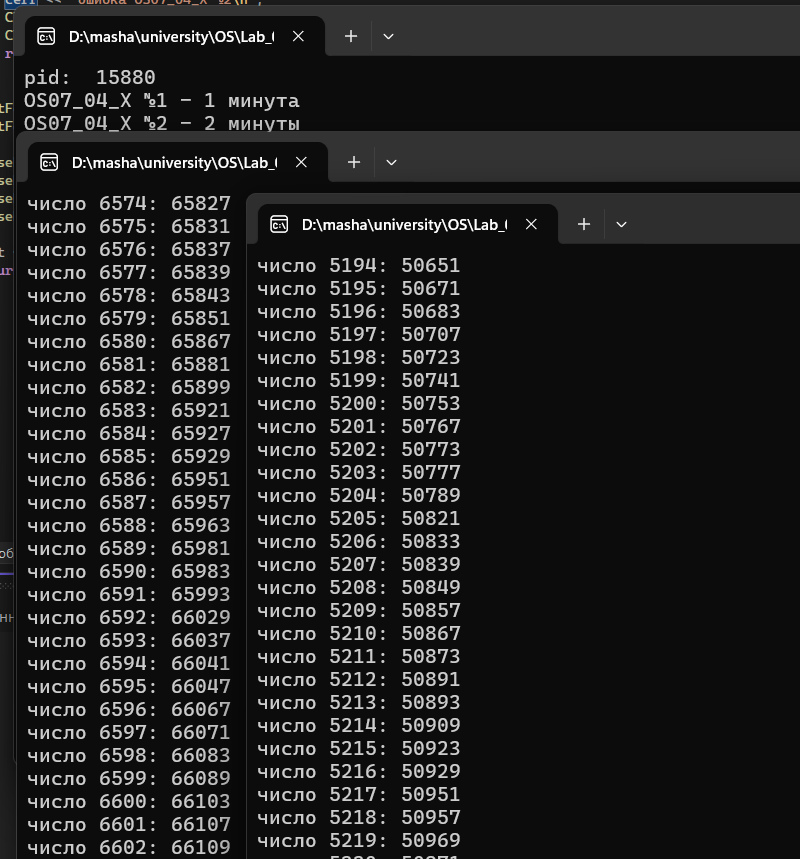
Первый дочерний процесс должен выполняться 1 минуту и корректно завершаться.

Первый дочерний процесс должен выполняться 2 минуты и корректно завершаться.

Приложение **OS07\_04** завершается после завершения дочерних процессов.

|  |
| --- |
| #include <Windows.h>  #include <iostream>  #include <ctime>  using namespace std;  int main() {  setlocale(LC\_ALL, "rus");  DWORD pid = GetCurrentProcessId();  cout << "pid: " << pid << '\n';  LPCWSTR path = L"D:\\masha\\university\\OS\\Lab\_07\\OS07\_04\_X\\x64\\Debug\\OS07\_04\_X.exe";  PROCESS\_INFORMATION pi1, pi2;  STARTUPINFO si;  ZeroMemory(&si, sizeof(si));  si.cb = sizeof(si);  if (CreateProcess(path, (LPWSTR)L" 1", NULL, NULL, FALSE, CREATE\_NEW\_CONSOLE, NULL, NULL, &si, &pi1)) {  cout << "OS07\_04\_X №1 - 1 минута\n";  }  else {  cerr << "ошибка OS07\_04\_X №1\n";  return 1;  }  if (CreateProcess(path, (LPWSTR)L" 2", NULL, NULL, FALSE, CREATE\_NEW\_CONSOLE, NULL, NULL, &si, &pi2)) {  cout << "OS07\_04\_X №2 - 2 минуты\n";  }  else {  cerr << "ошибка OS07\_04\_X №2\n";  CloseHandle(pi1.hProcess);  CloseHandle(pi1.hThread);  return 1;  }  WaitForSingleObject(pi1.hProcess, INFINITE);  WaitForSingleObject(pi2.hProcess, INFINITE);  CloseHandle(pi1.hProcess);  CloseHandle(pi1.hThread);  CloseHandle(pi2.hProcess);  CloseHandle(pi2.hThread);  cout << "конец\n";  return 0;  } |

|  |
| --- |
| #include <iostream>  #include <windows.h>  #include <ctime>  using namespace std;  bool isPrime(int num) {  if (num <= 1) return false;  for (int i = 2; i \* i <= num; i++) {  if (num % i == 0) return false;  }  return true;  }  int main(int argc, char\* argv[]) {  setlocale(LC\_ALL, "rus");  if (argc != 2) {  cerr << "дургое количество агрументов должнло быть" << endl;  return 1;  }  int duration = atoi(argv[1]);  DWORD startTime = GetTickCount();  int count = 0;  for (int num = 2; ; num++) {  if (isPrime(num)) {  cout << "число " << ++count << ": " << num << endl;  }  if ((GetTickCount() - startTime) / 1000 >= duration \* 60) {  cout << "процесс завершен через " << duration << " минут(ы)." << endl;  break;  }  }  return 0;  } |

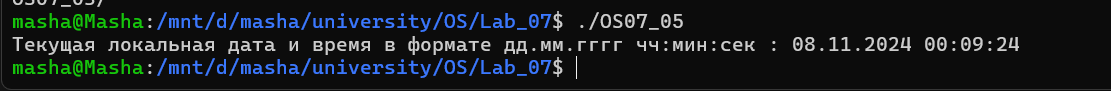


**Задание 05. Linux**

Разработайте приложение **OS07\_05**.

Приложение **OS07\_05** выводит на консоль текущую локальную дату и время в формате ***дд.мм.ггг чч:мин:сек***.

|  |
| --- |
| #include <stdio.h>  #include <time.h>  int main() {      time\_t currentTime = time(NULL);      struct tm\* localTimeStruct = localtime(&currentTime);      char timeBuffer[20];      strftime(timeBuffer, sizeof(timeBuffer), "%d.%m.%Y %H:%M:%S", localTimeStruct);      printf("Текущая локальная дата и время в формате дд.мм.гггг чч:мин:сек : %s\n", timeBuffer);      return 0;  } |



**Задание 06. Linux**

Разработайте приложение **OS07\_06,** выполняющее бесконечный цикл.

В теле цикла подсчитывается количество итераций.

Выведите на консоль значения счетчика итераций через 2 сек. **процессорного** времени и корректно завершите цикл.

Приложение **OS07\_06 должно** выполнять замер реального затраченного на работу цикла времени и выводить его значения на консоль.

|  |
| --- |
| #include <stdio.h>  #include <time.h>  #include <unistd.h>  int main() {      clock\_t startTime = clock();      clock\_t lastTime = startTime;      int iterationCount = 0;        time\_t realStartTime = time(NULL);        while (1) {          iterationCount++;            if ((clock() - lastTime) / CLOCKS\_PER\_SEC >= 2) {              lastTime = clock();              printf("Количество итераций через 2 секунды процессорного времени: %d\n", iterationCount);          }          time\_t currentRealTime = time(NULL);          if (currentRealTime - realStartTime >= 10) {              break;          }      }      clock\_t endTime = clock();      double cpuTimeUsed = ((double)(endTime - startTime)) / CLOCKS\_PER\_SEC;      printf("Общее процессорное время: %.2f секунд\n", cpuTimeUsed);      return 0;  } |

