Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Операционные системы

Студент: Коршун Н.И.

ФИТ 3 курс 5 группа

Преподаватель: Савельева М.Г.

Минск 2023

**Лабораторная работа №4**

Задание 1. Windows-приложение OS04\_01 на языке С++, выполняющее длинный цикл с временной задержкой и с выводом на консоль идентификаторов текущего процесса и текущего потока.

|  |  |
| --- | --- |
| Код программы OS04\_01 | #include <iostream>  #include <windows.h>  void DisplayProcessAndThreadInfo() {  DWORD processId = GetCurrentProcessId();  DWORD threadId = GetCurrentThreadId();  std::cout << "Process ID: " << processId << std::endl;  std::cout << "Thread ID: " << threadId << std::endl;  std::cout << "------------------------" << std::endl;  }  int main() {  while (true) {  DisplayProcessAndThreadInfo();  Sleep(1000);  }  return 0;  } |
| Скриншот запуска |  |
| Просмотр через Powershell |  |
| Просмотр через Perfomance Monitor |  |

Задание 2. Windows-приложение OS04\_02 на языке С++, выполняющее цикл 100 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса.

Процесс OS04\_02 должен создать два потока: потоковые функции OS04\_02\_T1, OS04\_02\_T2.

Поток OS04\_02\_T1 - выполняет цикл 50 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса и потока.

Поток OS04\_02\_T2 - выполняет цикл 125 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса и потока.

|  |  |
| --- | --- |
| Код программы OS04\_02 | #include <iostream>  #include <windows.h>  #include <thread>  DWORD WINAPI OS04\_02\_T1(LPVOID lpParam) {  for (int i = 0; i < 50; ++i) {  DWORD processId = GetCurrentProcessId();  DWORD threadId = GetCurrentThreadId();  std::cout << "Thread: OS04\_02\_T1, ";  std::cout << "Process ID: " << processId << ", ";  std::cout << "Thread ID: " << threadId << std::endl;  Sleep(1000);  }  return 0;  }  DWORD WINAPI OS04\_02\_T2(LPVOID lpParam) {  for (int i = 0; i < 125; ++i) {  DWORD processId = GetCurrentProcessId();  DWORD threadId = GetCurrentThreadId();  std::cout << "Thread: OS04\_02\_T2, ";  std::cout << "Process ID: " << processId << ", ";  std::cout << "Thread ID: " << threadId << std::endl;  Sleep(1000);  }  return 0;  }  int main() {  HANDLE thread1 = CreateThread(NULL, 0, OS04\_02\_T1, NULL, 0, NULL);  HANDLE thread2 = CreateThread(NULL, 0, OS04\_02\_T2, NULL, 0, NULL);  for (int i = 0; i < 100; ++i) {  DWORD processId = GetCurrentProcessId();  DWORD threadId = GetCurrentThreadId();  std::cout << "Thread: OS04\_02, ";  std::cout << "Process ID: " << processId << ", ";  std::cout << "Thread ID: " << threadId << std::endl;  std::this\_thread::sleep\_for(std::chrono::seconds(1));  }  // Ожидание завершения потоков  WaitForSingleObject(thread1, INFINITE);  WaitForSingleObject(thread2, INFINITE);  // Закрытие хендлов потоков  CloseHandle(thread1);  CloseHandle(thread2);  return 0;  } |
| Скриншот запуска |  |
| Просмотр через Powershell |  |
| Просмотр через Perfomance Monitor |  |

Задание 3. Windows-приложение OS04\_03 на языке С++, выполняющее цикл 100 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса.

Процесс OS04\_03 должен создать два потока: потоковые функции OS04\_03\_T1, OS04\_03\_T2.

Поток OS04\_03\_T1 - выполняет цикл 50 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса и потока.

Поток OS04\_03\_T2 - выполняет цикл 125 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса и потока.

Поток main приостанавливает работу потока OS04\_03\_T1 на 20й своей итерации и возобновляют на 60й своей итерации.

Поток main приостанавливает работу потока OS04\_03\_T2 на 40й своей итерации и возобновляют после окончания собственных итераций (по окончании собственного цикла).

|  |  |
| --- | --- |
| Код программы OS04\_03 | #include <iostream>  #include <windows.h>  DWORD WINAPI OS04\_03\_T1(LPVOID lpParam) {  DWORD processId = GetCurrentProcessId();  DWORD threadId = GetCurrentThreadId();  for (int i = 0; i < 50; ++i) {  std::cout << "Thread: OS04\_03\_T1 ";  std::cout << "Process ID: " << processId << ", ";  std::cout << "Thread ID: " << threadId << std::endl << std::endl;  Sleep(1000);  if (i == 19) {  std::cout << "Main thread is pausing OS04\_03\_T1 at iteration 20" << std::endl;  Sleep(20000);  std::cout << "Main thread is resuming OS04\_03\_T1 at iteration 21" << std::endl;  }  }  return 0;  }  DWORD WINAPI OS04\_03\_T2(LPVOID lpParam) {  DWORD processId = GetCurrentProcessId();  DWORD threadId = GetCurrentThreadId();  for (int i = 0; i < 125; ++i) {  std::cout << "Thread: OS04\_03\_T1, ";  std::cout << "Process ID: " << processId << ", ";  std::cout << "Thread ID: " << threadId << std::endl << std::endl;  Sleep(1000);  if (i == 39) {  std::cout << "Main thread is pausing OS04\_03\_T2 at iteration 40" << std::endl;  Sleep(20000);  std::cout << "Main thread is resuming OS04\_03\_T2 after its completion" << std::endl;  }  }  return 0;  }  int main() {  HANDLE thread1 = CreateThread(NULL, 0, OS04\_03\_T1, NULL, 0, NULL);  HANDLE thread2 = CreateThread(NULL, 0, OS04\_03\_T2, NULL, 0, NULL);  DWORD processId = GetCurrentProcessId();  DWORD threadId = GetCurrentThreadId();  for (int i = 0; i < 100; ++i) {  std::cout << "Thread: OS04\_03, ";  std::cout << "Process ID: " << processId << ", ";  std::cout << "Thread ID: " << threadId << std::endl << std::endl;  Sleep(1000);  if (i == 59) {  std::cout << "Main thread is resuming OS04\_03\_T1 at iteration 60" << std::endl;  WaitForSingleObject(thread1, INFINITE);  }  }  WaitForSingleObject(thread2, INFINITE);  CloseHandle(thread1);  CloseHandle(thread2);  return 0;  } |
| Скриншот запуска |  |
| Просмотр через Powershell |  |
| Просмотр через Perfomance Monitor |  |

Задание 4. Windows-приложение OS04\_04 на языке С++, выполняющее цикл 100 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса.

Процесс OS04\_04 должен создать два потока: потоковые функции OS04\_04\_T1, OS04\_04\_T2.

Поток OS04\_04\_T1 - выполняет цикл 50 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса и потока. Поток засыпает на 25й собственный итерации на 10сек.

Поток OS04\_04\_T2 - выполняет цикл 125 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса и потока. Поток засыпает на 80й собственной итерации на 15сек.

Поток main засыпает на 30й итерации на 10 сек.

|  |  |
| --- | --- |
| Код программы OS04\_04 | #include <iostream>  #include <windows.h>  DWORD WINAPI OS04\_04\_T1(LPVOID lpParam) {  for (int i = 0; i < 50; ++i) {  DWORD processId = GetCurrentProcessId();  DWORD threadId = GetCurrentThreadId();  std::cout << "Thread: OS04\_04\_T1, ";  std::cout << "Process ID: " << processId << ", ";  std::cout << "Thread ID: " << threadId << std::endl;  Sleep(1000);  if (i == 24) {  std::cout << "OS04\_04\_T1 is sleeping for 10 seconds at iteration 25" << std::endl;  Sleep(10000);  std::cout << "OS04\_04\_T1 has awakened at iteration 26" << std::endl;  }  }  return 0;  }  DWORD WINAPI OS04\_04\_T2(LPVOID lpParam) {  for (int i = 0; i < 125; ++i) {  DWORD processId = GetCurrentProcessId();  DWORD threadId = GetCurrentThreadId();  std::cout << "Thread: OS04\_04\_T2, ";  std::cout << "Process ID: " << processId << ", ";  std::cout << "Thread ID: " << threadId << std::endl;  Sleep(1000);  if (i == 79) {  std::cout << "OS04\_04\_T2 is sleeping for 15 seconds at iteration 80" << std::endl;  Sleep(15000);  std::cout << "OS04\_04\_T2 has awakened at iteration 81" << std::endl;  }  }  return 0;  }  int main() {  HANDLE thread1 = CreateThread(NULL, 0, OS04\_04\_T1, NULL, 0, NULL);  HANDLE thread2 = CreateThread(NULL, 0, OS04\_04\_T2, NULL, 0, NULL);  DWORD processId = GetCurrentProcessId();  DWORD threadId = GetCurrentThreadId();  for (int i = 0; i < 100; ++i) {  DWORD processId = GetCurrentProcessId();  DWORD threadId = GetCurrentThreadId();  std::cout << "Thread: OS04\_04, ";  std::cout << "Process ID: " << processId << ", ";  std::cout << "Thread ID: " << threadId << std::endl;  Sleep(1000);  if (i == 29) {  std::cout << "Main thread is sleeping for 10 seconds at iteration 30" << std::endl;  Sleep(10000);  std::cout << "Main thread has awakened at iteration 31" << std::endl;  }  }  WaitForSingleObject(thread1, INFINITE);  WaitForSingleObject(thread2, INFINITE);  CloseHandle(thread1);  CloseHandle(thread2);  return 0;  } |
| Скриншот запуска |  |
| Просмотр через Powershell |  |
| Просмотр через Perfomance Monitor |  |

Задание 5. Windows-приложение OS04\_05 на языке С++, выполняющее цикл 100 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса.

Процесс OS04\_05 должен создать два потока: потоковые функции OS04\_05\_T1, OS04\_05\_T2.

Поток OS04\_05\_T1 - выполняет цикл 50 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса и потока.

Поток OS04\_05\_T2 - выполняет цикл 125 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса и потока.

Поток main завершает поток OS04\_05\_T2 на 40й собственной итерации.

|  |  |
| --- | --- |
| Код программы OS04\_05 | #include <iostream>  #include <windows.h>  DWORD WINAPI OS04\_05\_T1(LPVOID lpParam) {  for (int i = 0; i < 50; ++i) {  std::cout << i + 1 << ") ";  DWORD processId = GetCurrentProcessId();  DWORD threadId = GetCurrentThreadId();  std::cout << "Thread: OS04\_05\_T1, ";  std::cout << "Process ID: " << processId << ", ";  std::cout << "Thread ID: " << threadId << std::endl;  Sleep(1000);  }  return 0;  }  DWORD WINAPI OS04\_05\_T2(LPVOID lpParam) {  for (int i = 0; i < 125; ++i) {  std::cout << i + 1 << ") ";  DWORD processId = GetCurrentProcessId();  DWORD threadId = GetCurrentThreadId();  std::cout << "Thread: OS04\_05\_T1, ";  std::cout << "Process ID: " << processId << ", ";  std::cout << "Thread ID: " << threadId << std::endl;  Sleep(1000);  if (i == 39) {  std::cout << "Main thread is terminating OS04\_05\_T2 at iteration 40" << std::endl;  return 0;  }  }  return 0;  }  int main() {  HANDLE thread1 = CreateThread(NULL, 0, OS04\_05\_T1, NULL, 0, NULL);  HANDLE thread2 = CreateThread(NULL, 0, OS04\_05\_T2, NULL, 0, NULL);  DWORD processId = GetCurrentProcessId();  DWORD threadId = GetCurrentThreadId();  for (int i = 0; i < 100; ++i) {  std::cout << i + 1 << ") ";  DWORD processId = GetCurrentProcessId();  DWORD threadId = GetCurrentThreadId();  std::cout << "Thread: OS04\_05\_T1, ";  std::cout << "Process ID: " << processId << ", ";  std::cout << "Thread ID: " << threadId << std::endl;  Sleep(1000);  }  WaitForSingleObject(thread1, INFINITE);  WaitForSingleObject(thread2, INFINITE);  CloseHandle(thread1);  CloseHandle(thread2);  return 0;  } |
| Скриншот запуска |  |
| Просмотр через Powershell |  |
| Просмотр через Perfomance Monitor |  |

Задание 6. Linux-приложение OS04\_06 на языке С, выполняющее длинный цикл с временной задержкой и с выводом на консоль идентификатора процесса.

|  |  |
| --- | --- |
| Код программы OS04\_06 | #include <stdio.h>  #include <unistd.h>  int main() {  for (int i = 0; i < 100; ++i) {  printf("Process ID: %d\n", getpid());  fflush(stdout); // Очистка буфера вывода  // Временная задержка в секундах  sleep(1);  }  return 0;  } |
| Скриншот запуска |  |

Задание 7. Linux-приложение OS04\_07 на языке С, выполняющее цикл 100 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса.

Процесс OS04\_07 должен создать поток: потоковая функция OS04\_07\_T1.

Поток OS04\_07\_T1 - выполняет цикл 75 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса.

|  |  |
| --- | --- |
| Код программы OS04\_07 | #include <stdio.h>  #include <unistd.h>  #include <pthread.h>  // Структура для передачи данных в поток  struct ThreadData {  pid\_t processId;  };  // Потоковая функция  void\* OS04\_07\_T1(void\* arg) {  struct ThreadData\* data = (struct ThreadData\*)arg;  for (int i = 0; i < 75; ++i) {  printf("Process ID: %d, Thread ID: %lu\n", data->processId, pthread\_self());  fflush(stdout); // Очистка буфера вывода  // Временная задержка в секундах  sleep(1);  }  return NULL;  }  int main() {  pid\_t processId = getpid();  printf("Process ID: %d\n", processId);  // Инициализация структуры с данными для передачи в поток  struct ThreadData data;  data.processId = processId;  // Создание потока  pthread\_t thread;  if (pthread\_create(&thread, NULL, OS04\_07\_T1, (void\*)&data) != 0) {  fprintf(stderr, "Error creating thread\n");  return 1;  }  // Ожидание завершения потока  if (pthread\_join(thread, NULL) != 0) {  fprintf(stderr, "Error joining thread\n");  return 1;  }  return 0;  } |
| Скриншот запуска |  |

Задание 8. Linux-приложение OS04\_08 на языке С, выполняющее цикл 100 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса.

Процесс OS04\_08 должен создать поток: потоковая функция OS04\_08\_T1.

Поток OS04\_08\_T1 - выполняет цикл 75 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса. Поток должен засыпать на 10сек. на 50й собственной итерации.

Поток main должен засыпать на 15 сек. на 30й собственной итерации.

|  |  |
| --- | --- |
| Код программы OS04\_08 | #include <stdio.h>  #include <unistd.h>  #include <pthread.h>  // Структура для передачи данных в поток  struct ThreadData {  pid\_t processId;  };  // Потоковая функция  void\* OS04\_08\_T1(void\* arg) {  struct ThreadData\* data = (struct ThreadData\*)arg;  for (int i = 0; i < 75; ++i) {  printf("Process ID: %d, Thread ID: %lu\n", data->processId, pthread\_self());  fflush(stdout); // Очистка буфера вывода  // Проверка на 50-ю итерацию  if (i == 49) {  printf("Thread OS04\_08\_T1 is sleeping for 10 seconds at iteration 50\n");  fflush(stdout);  sleep(10);  printf("Thread OS04\_08\_T1 has awakened at iteration 51\n");  fflush(stdout);  }  // Временная задержка в секундах  sleep(1);  }  return NULL;  }  int main() {  pid\_t processId = getpid();  printf("Process ID: %d\n", processId);  // Инициализация структуры с данными для передачи в поток  struct ThreadData data;  data.processId = processId;  for (int i = 0; i < 100; ++i) {  printf("Main thread iteration: %d\n", i);  fflush(stdout);  // Проверка на 30-ю итерацию  if (i == 29) {  printf("Main thread is sleeping for 15 seconds at iteration 30\n");  fflush(stdout);  sleep(15);  printf("Main thread has awakened at iteration 31\n");  fflush(stdout);  }  // Временная задержка в секундах  sleep(1);  }  // Создание потока  pthread\_t thread;  if (pthread\_create(&thread, NULL, OS04\_08\_T1, (void\*)&data) != 0) {  fprintf(stderr, "Error creating thread\n");  return 1;  }  // Ожидание завершения потока  if (pthread\_join(thread, NULL) != 0) {  fprintf(stderr, "Error joining thread\n");  return 1;  }  return 0;  } |
| Скриншот запуска |  |