Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Операционные системы

Студент: Песецкий Н.А.

ФИТ 3 курс 5 группа

Преподаватель: Савельева М.Г.

Минск 2023

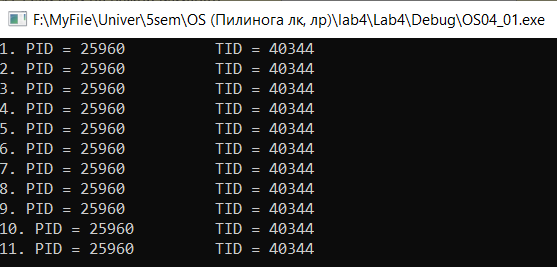
**Лабораторная работа №4**

**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.

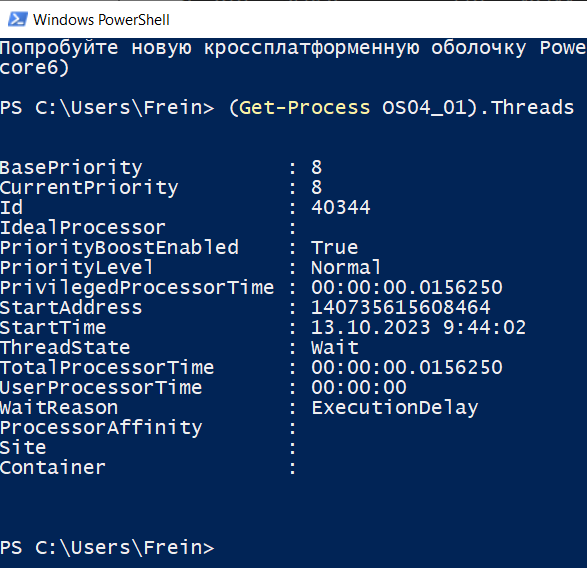
Код программы OS04\_01

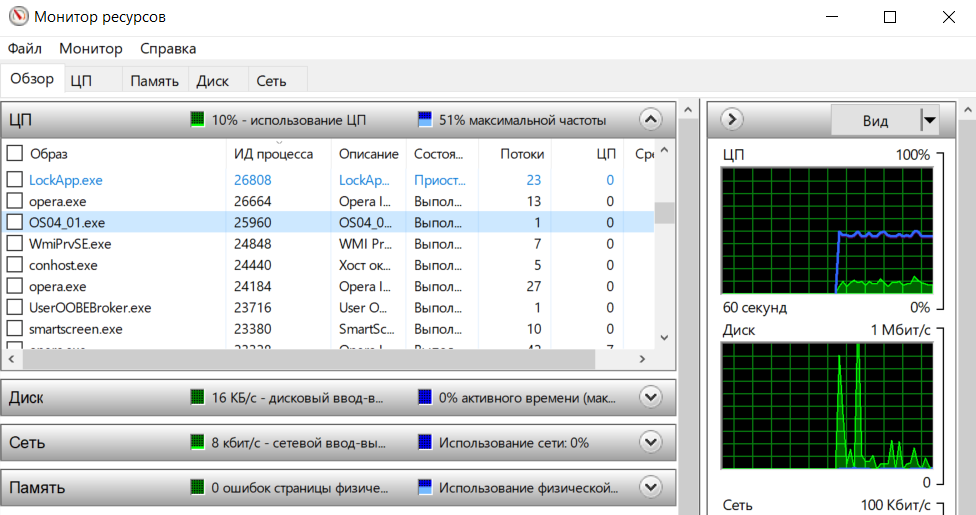
|  |
| --- |
| #include <iostream>  #include <Windows.h>  using namespace std;  int main()  {  DWORD pid = GetCurrentProcessId();  DWORD tid = GetCurrentThreadId();  for (short i = 1; i <= 10000; ++i)  {  cout << i << ". PID = " << pid << "\t\tTID = " << tid << "\n";  Sleep(1000);  }  system("pause");  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.

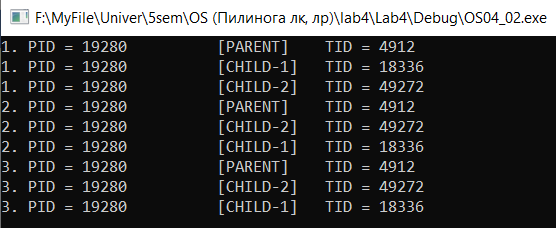
Поток OS04\_02\_T1 - выполняет цикл 50 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса и потока.

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

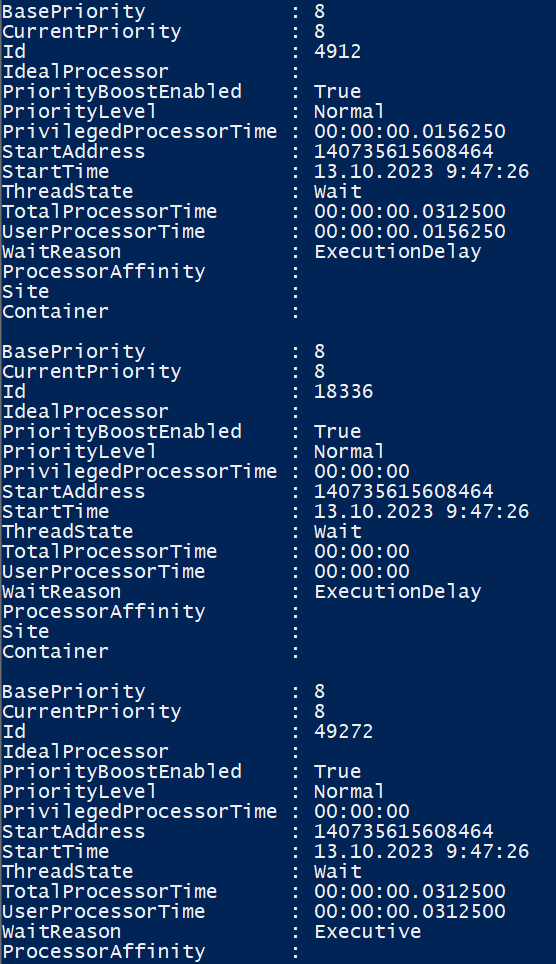
Код программы OS04\_02

|  |
| --- |
| #include <iostream>  #include <Windows.h>  using namespace std;  DWORD pid = NULL;  DWORD WINAPI ChildThread\_T1()  {  DWORD tid = GetCurrentThreadId();  for (short i = 1; i <= 50; ++i)  {  cout << i << ". PID = " << pid << "\t\t[CHILD-1] TID = " << tid << "\n";  Sleep(1000);  }  return 0;  }  DWORD WINAPI ChildThread\_T2()  {  DWORD tid = GetCurrentThreadId();  for (short i = 1; i <= 125; ++i)  {  cout << i << ". PID = " << pid << "\t\t[CHILD-2] TID = " << tid << "\n";  Sleep(1000);  }  return 0;  }  // Powershell ISE: (Get-Process OS04\_02).Threads  int main()  {  pid = GetCurrentProcessId();  DWORD parentId = GetCurrentThreadId();  DWORD childId\_T1 = NULL;  DWORD childId\_T2 = NULL;  HANDLE handleClild\_T1 = CreateThread(NULL, 0, (LPTHREAD\_START\_ROUTINE)ChildThread\_T1, NULL, 0, &childId\_T1);  HANDLE handleClild\_T2 = CreateThread(NULL, 0, (LPTHREAD\_START\_ROUTINE)ChildThread\_T2, NULL, 0, &childId\_T2);  for (short i = 1; i <= 100; ++i)  {  cout << i << ". PID = " << pid << "\t\t[PARENT] TID = " << parentId << "\n";  Sleep(1000);  }  WaitForSingleObject(handleClild\_T1, INFINITE);  WaitForSingleObject(handleClild\_T2, INFINITE);  CloseHandle(handleClild\_T1);  CloseHandle(handleClild\_T2);  system("pause");  return 0;  } |

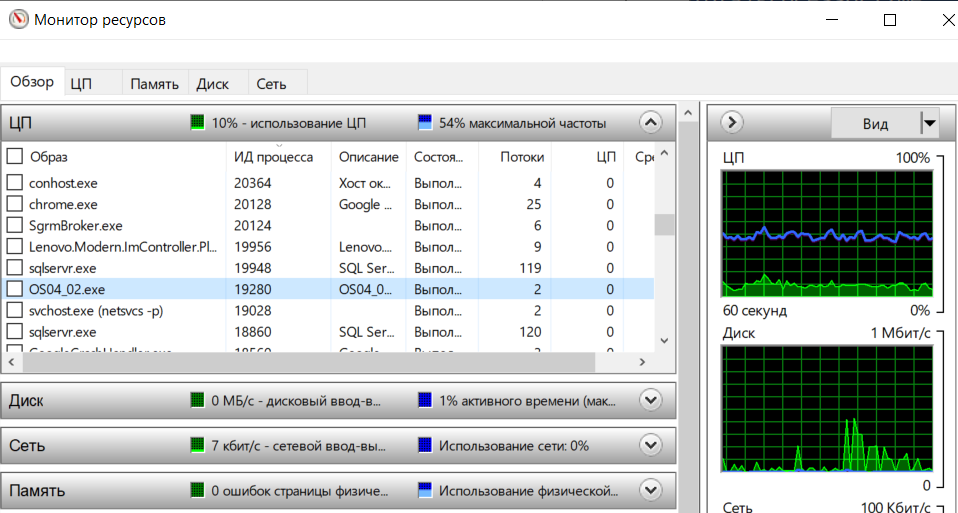
Скриншот запуска



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.

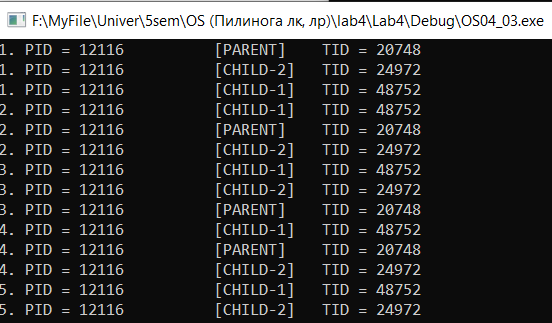
Поток main приостанавливает работу потока OS04\_03\_T1 на 20й своей итерации и возобновляют на 60й своей итерации.

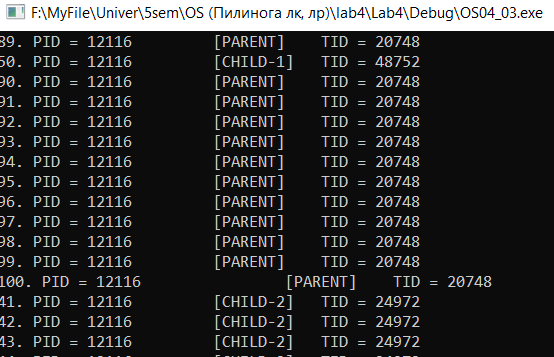
Поток main приостанавливает работу потока OS04\_03\_T2 на 40й своей итерации и возобновляют после окончания собственных итераций (по окончании собственного цикла).

Код программы OS04\_03

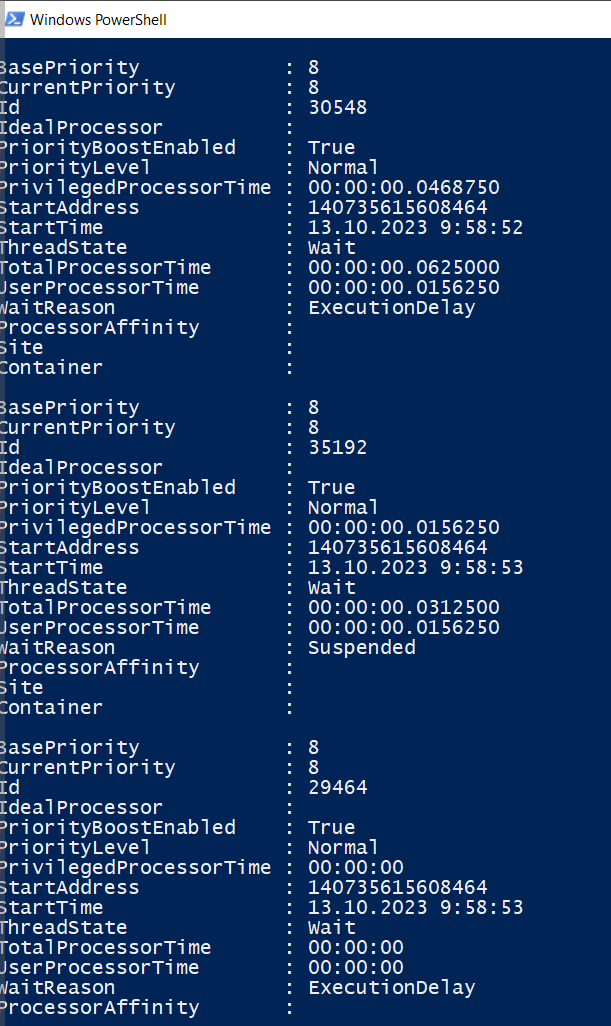
|  |
| --- |
| #include <iostream>  #include <Windows.h>  using namespace std;  DWORD pid = NULL;  DWORD WINAPI ChildThread\_T1()  {  DWORD tid = GetCurrentThreadId();  for (short i = 1; i <= 50; ++i)  {  cout << i << ". PID = " << pid << "\t\t[CHILD-1] TID = " << tid << "\n";  Sleep(1000);  }  return 0;  }  DWORD WINAPI ChildThread\_T2()  {  DWORD tid = GetCurrentThreadId();  for (short i = 1; i <= 125; ++i)  {  cout << i << ". PID = " << pid << "\t\t[CHILD-2] TID = " << tid << "\n";  Sleep(1000);  }  return 0;  }  // Powershell ISE: (Get-Process OS04\_03).Threads  int main()  {  pid = GetCurrentProcessId();  DWORD parentId = GetCurrentThreadId();  DWORD childId\_T1 = NULL;  DWORD childId\_T2 = NULL;  HANDLE handleClild\_T1 = CreateThread(NULL, 0, (LPTHREAD\_START\_ROUTINE)ChildThread\_T1, NULL, 0, &childId\_T1);  HANDLE handleClild\_T2 = CreateThread(NULL, 0, (LPTHREAD\_START\_ROUTINE)ChildThread\_T2, NULL, 0, &childId\_T2);  for (short i = 1; i <= 100; ++i)  {  switch (i)  {  case 20: SuspendThread(handleClild\_T1); break;  case 60: ResumeThread(handleClild\_T1); break;  case 40: SuspendThread(handleClild\_T2); break;  case 100: ResumeThread(handleClild\_T2); break;  }  cout << i << ". PID = " << pid << "\t\t[PARENT] TID = " << parentId << "\n";  Sleep(1000);  }  WaitForSingleObject(handleClild\_T1, INFINITE);  WaitForSingleObject(handleClild\_T2, INFINITE);  CloseHandle(handleClild\_T1);  CloseHandle(handleClild\_T2);  system("pause");  return 0;  } |

Скриншот запуска

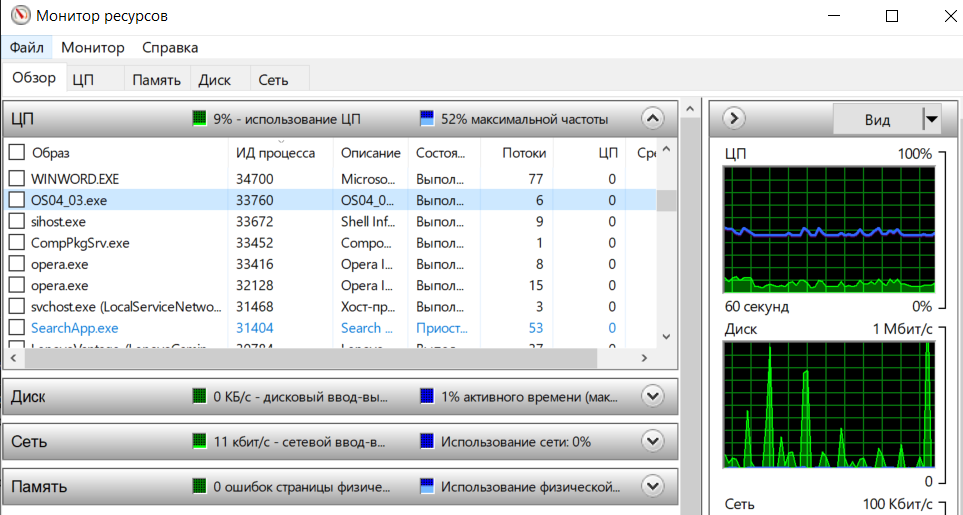




Просмотр через Powershell



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.

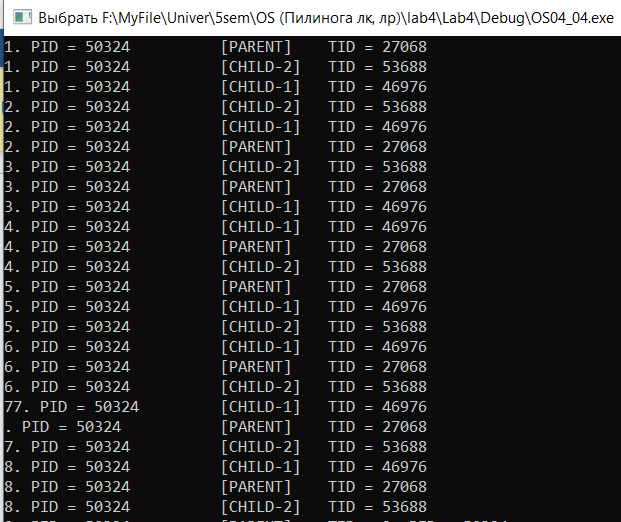
Поток OS04\_04\_T2 - выполняет цикл 125 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса и потока. Поток засыпает на 80й собственной итерации на 15сек.

Поток main засыпает на 30й итерации на 10 сек.

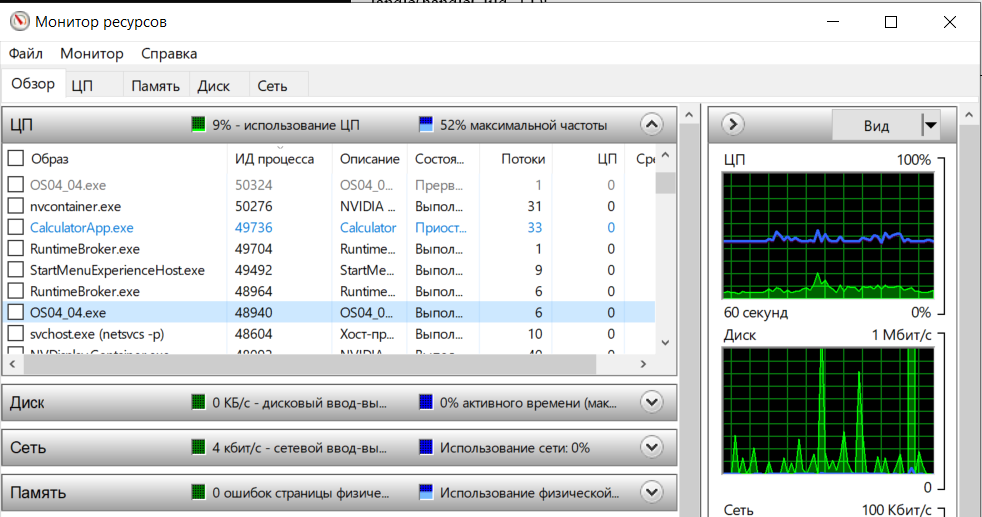
Код программы OS04\_04

|  |
| --- |
| #include <iostream>  #include <Windows.h>  using namespace std;  DWORD pid = NULL;  DWORD WINAPI ChildThread\_T1(){  DWORD tid = GetCurrentThreadId();  for (short i = 1; i <= 50; ++i)  {  cout << i << ". PID = " << pid << "\t\t[CHILD-1] TID = " << tid << "\n";  Sleep(1000);  if (i == 25)  Sleep(10000);  }  return 0;  }  DWORD WINAPI ChildThread\_T2(){  DWORD tid = GetCurrentThreadId();  for (short i = 1; i <= 125; ++i)  {  cout << i << ". PID = " << pid << "\t\t[CHILD-2] TID = " << tid << "\n";  Sleep(1000);  if (i == 80)  Sleep(15000);  }  return 0;  }  // Powershell ISE: (Get-Process OS04\_04).Threads  int main(){  pid = GetCurrentProcessId();  DWORD parentId = GetCurrentThreadId();  DWORD childId\_T1 = NULL;  DWORD childId\_T2 = NULL;  HANDLE handleClild\_T1 = CreateThread(NULL, 0, (LPTHREAD\_START\_ROUTINE)ChildThread\_T1, NULL, NULL, &childId\_T1);  HANDLE handleClild\_T2 = CreateThread(NULL, 0, (LPTHREAD\_START\_ROUTINE)ChildThread\_T2, NULL, NULL, &childId\_T2);  for (short i = 1; i <= 100; ++i)  {  if (i == 30)  Sleep(10000);  cout << i << ". PID = " << pid << "\t\t[PARENT] TID = " << parentId << "\n";  Sleep(1000);  }  WaitForSingleObject(handleClild\_T1, INFINITE);  WaitForSingleObject(handleClild\_T2, INFINITE);  CloseHandle(handleClild\_T1);  CloseHandle(handleClild\_T2);  system("pause");  return 0;  } |

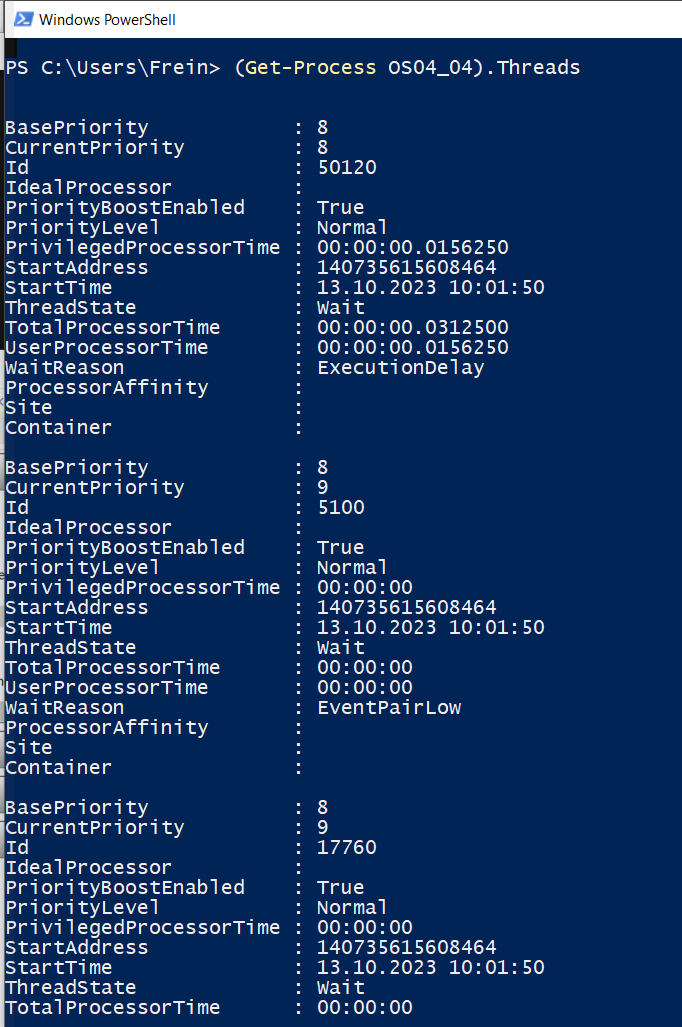
Скриншот запуска

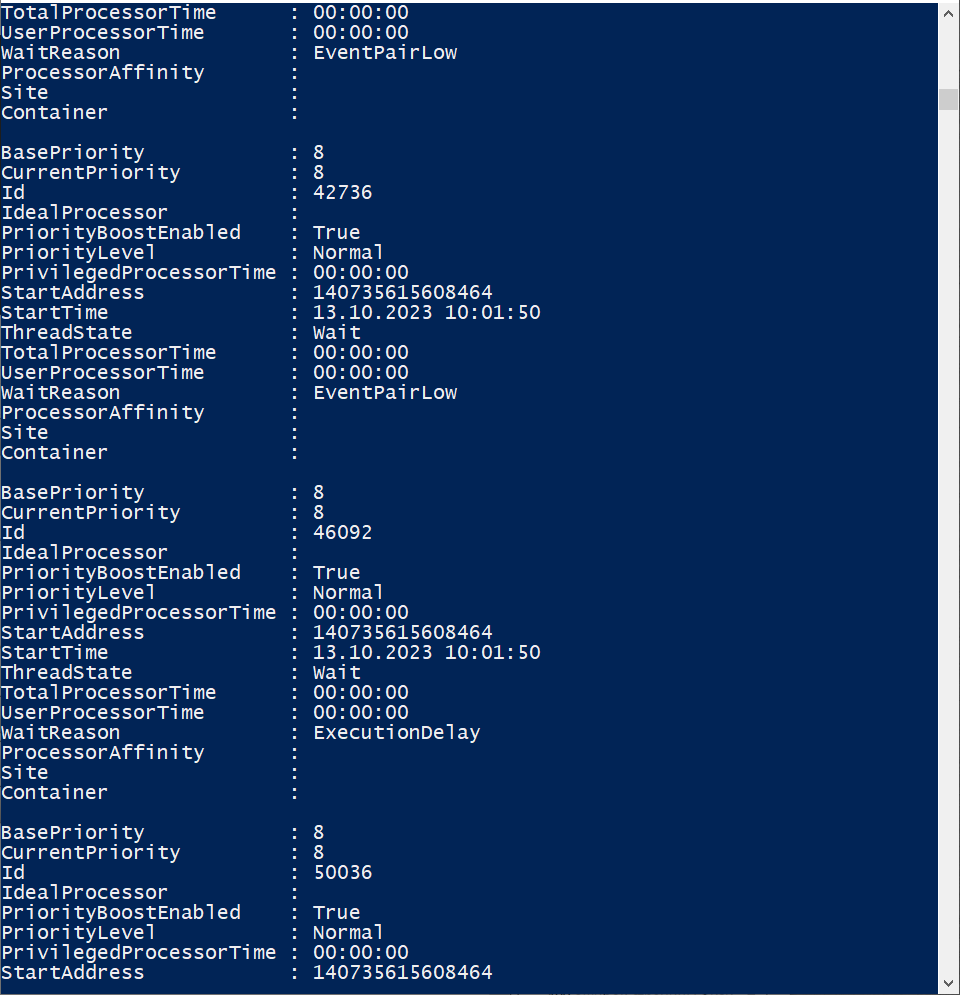


Просмотр через Perfomance Monitor



Просмотр через Powershell





**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.

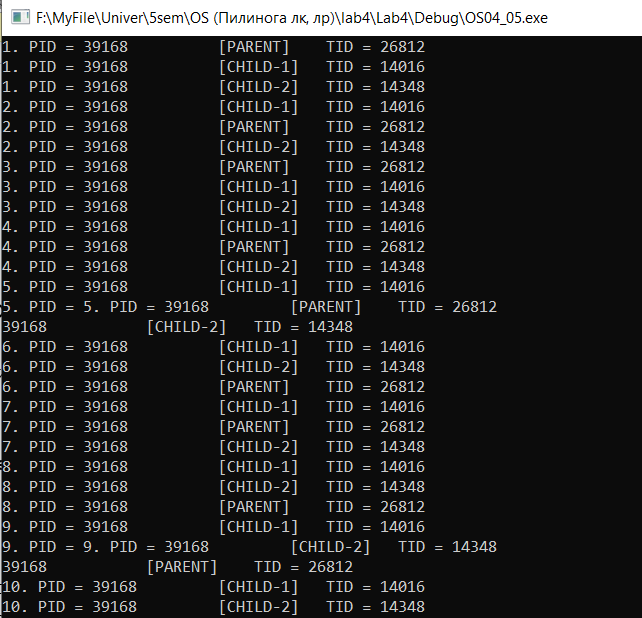
Поток OS04\_05\_T2 - выполняет цикл 125 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса и потока.

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

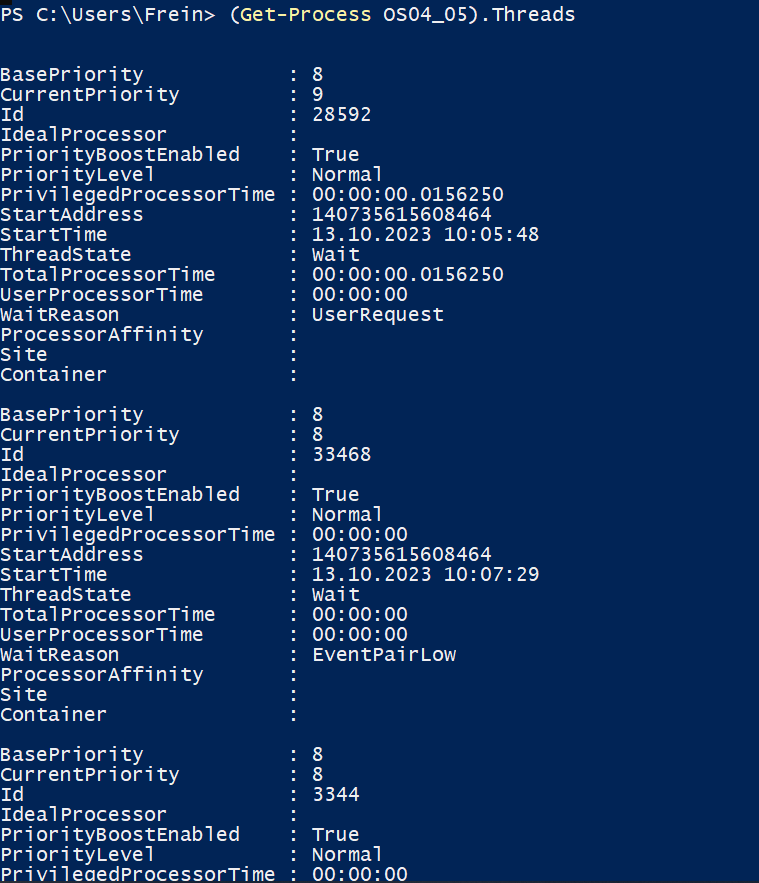
Код программы OS04\_05

|  |
| --- |
| #include <iostream>  #include <Windows.h>  using namespace std;  DWORD pid = NULL;  DWORD WINAPI ChildThread\_T1(){  DWORD tid = GetCurrentThreadId();  for (short i = 1; i <= 50; ++i)  {  cout << i << ". PID = " << pid << "\t\t[CHILD-1] TID = " << tid << "\n";  Sleep(1000);  }  return 0;  }  DWORD WINAPI ChildThread\_T2(){  DWORD tid = GetCurrentThreadId();  for (short i = 1; i <= 125; ++i)  {  cout << i << ". PID = " << pid << "\t\t[CHILD-2] TID = " << tid << "\n";  Sleep(1000);  }  return 0;  }  // Powershell ISE: (Get-Process OS04\_05).Threads  int main(){  pid = GetCurrentProcessId();  DWORD parentId = GetCurrentThreadId();  DWORD childId\_T1 = NULL;  DWORD childId\_T2 = NULL;  HANDLE handleClild\_T1 = CreateThread(NULL, 0, (LPTHREAD\_START\_ROUTINE)ChildThread\_T1, NULL, NULL, &childId\_T1);  HANDLE handleClild\_T2 = CreateThread(NULL, 0, (LPTHREAD\_START\_ROUTINE)ChildThread\_T2, NULL, NULL, &childId\_T2);  for (short i = 1; i <= 100; ++i) {  if (i == 40)  TerminateThread(handleClild\_T2, -1);  cout << i << ". PID = " << pid << "\t\t[PARENT] TID = " << parentId << "\n";  Sleep(1000);  }  WaitForSingleObject(handleClild\_T1, INFINITE);  WaitForSingleObject(handleClild\_T2, INFINITE);  CloseHandle(handleClild\_T1);  CloseHandle(handleClild\_T2);  system("pause");  return 0;} |

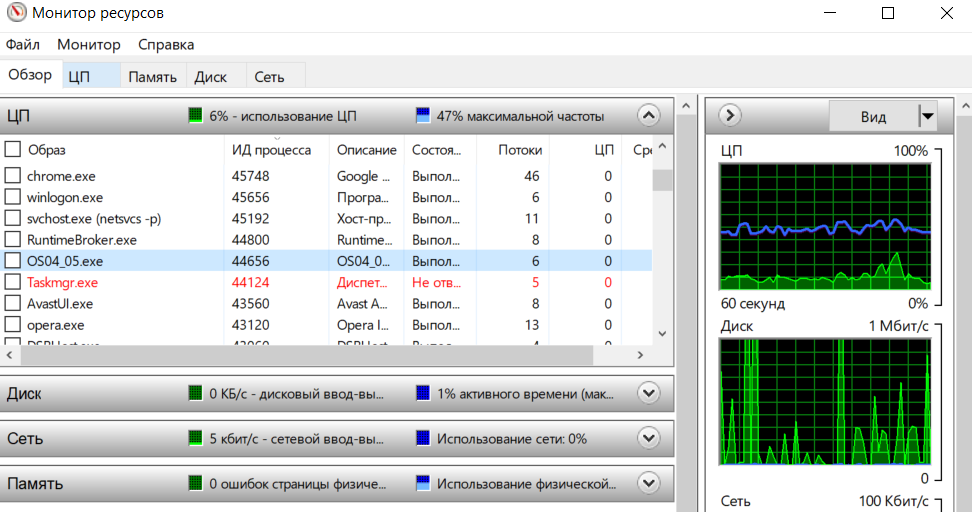
Скриншот запуска



Просмотр через Powershell



Просмотр через Perfomance Monitor



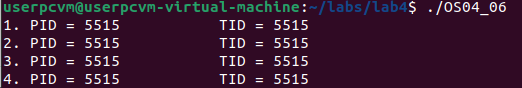
**Задание 6.** Linux-приложение OS04\_06 на языке С, выполняющее длинный цикл с временной задержкой и с выводом на консоль идентификатора процесса.

Код программы OS04\_06

|  |
| --- |
| #define \_GNU\_SOURCE  #include <stdio.h>  #include <stdlib.h>  #include <sys/types.h>  #include <unistd.h>  /\*  ps -Lp <PID> H  ps -fL -p <PID> H  \*/  int main()  {  pid\_t pid = getpid();  pid\_t tid = gettid();  for (short i = 1; i <= 10000; ++i)  {  printf("%d. PID = %d\t\tTID = %d\n", i, pid, tid);  sleep(1);  }  exit(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.

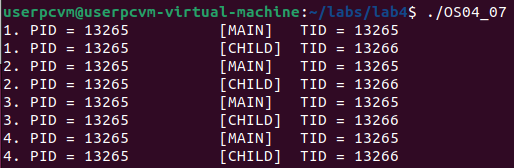
Процесс OS04\_07 должен создать поток: потоковая функция OS04\_07\_T1.

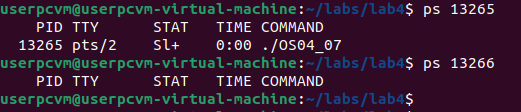
Поток OS04\_07\_T1 - выполняет цикл 75 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса.

Код программы OS04\_07

|  |
| --- |
| #define \_GNU\_SOURCE  #include <stdio.h>  #include <stdlib.h>  #include <unistd.h>  #include <pthread.h>  #include <sys/types.h>  #include <sys/syscall.h>  /\*  ps -Lp <PID> H  ps -fL -p <PID> H  \*/  void\* ChildThread\_1(void\* arg)  {  pid\_t pid = getpid();  pid\_t tid = syscall(SYS\_gettid);  for (int i = 1; i <= 75; ++i)  {  printf("%d. PID = %d\t\t[CHILD] TID = %d\n", i, pid, tid);  sleep(1);  }  pthread\_exit("Child thread");  }  int main()  {  pid\_t pid = getpid();  pid\_t tid = syscall(SYS\_gettid);  pthread\_t a\_thread;  void\* thread\_result;  int res = pthread\_create(&a\_thread, NULL, ChildThread\_1, NULL);  for (int i = 1; i <= 100; ++i)  {  printf("%d. PID = %d\t\t[MAIN] TID = %d\n", i, pid, tid);  sleep(1);  }  int status = pthread\_join(a\_thread, (void\*\*)&thread\_result);  exit(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

|  |
| --- |
| #define \_GNU\_SOURCE  #include <stdio.h>  #include <stdlib.h>  #include <unistd.h>  #include <pthread.h>  #include <sys/types.h>  #include <sys/syscall.h>  /\*  ps -Lp <PID> H  ps -fL -p <PID> H  \*/  void\* ChildThread\_1(void\* arg)  {  pid\_t pid = getpid();  pid\_t tid = syscall(SYS\_gettid);  for (int i = 1; i <= 75; ++i)  {  printf("%d. PID = %d\t\t[CHILD] TID = %d\n", i, pid, tid);  sleep(1);  if (i == 50)  sleep(10);  }  pthread\_exit("Child thread");  }  int main()  {  pid\_t pid = getpid();  pid\_t tid = syscall(SYS\_gettid);  pthread\_t a\_thread;  void\* thread\_result;  int res = pthread\_create(&a\_thread, NULL, ChildThread\_1, NULL);  for (int i = 1; i <= 100; ++i)  {  printf("%d. PID = %d\t\t[MAIN] TID = %d\n", i, pid, tid);  sleep(1);  if (i == 30)  sleep(15);  }  int status = pthread\_join(a\_thread, (void\*\*)&thread\_result);  exit(0);  } |

Скриншот запуска

