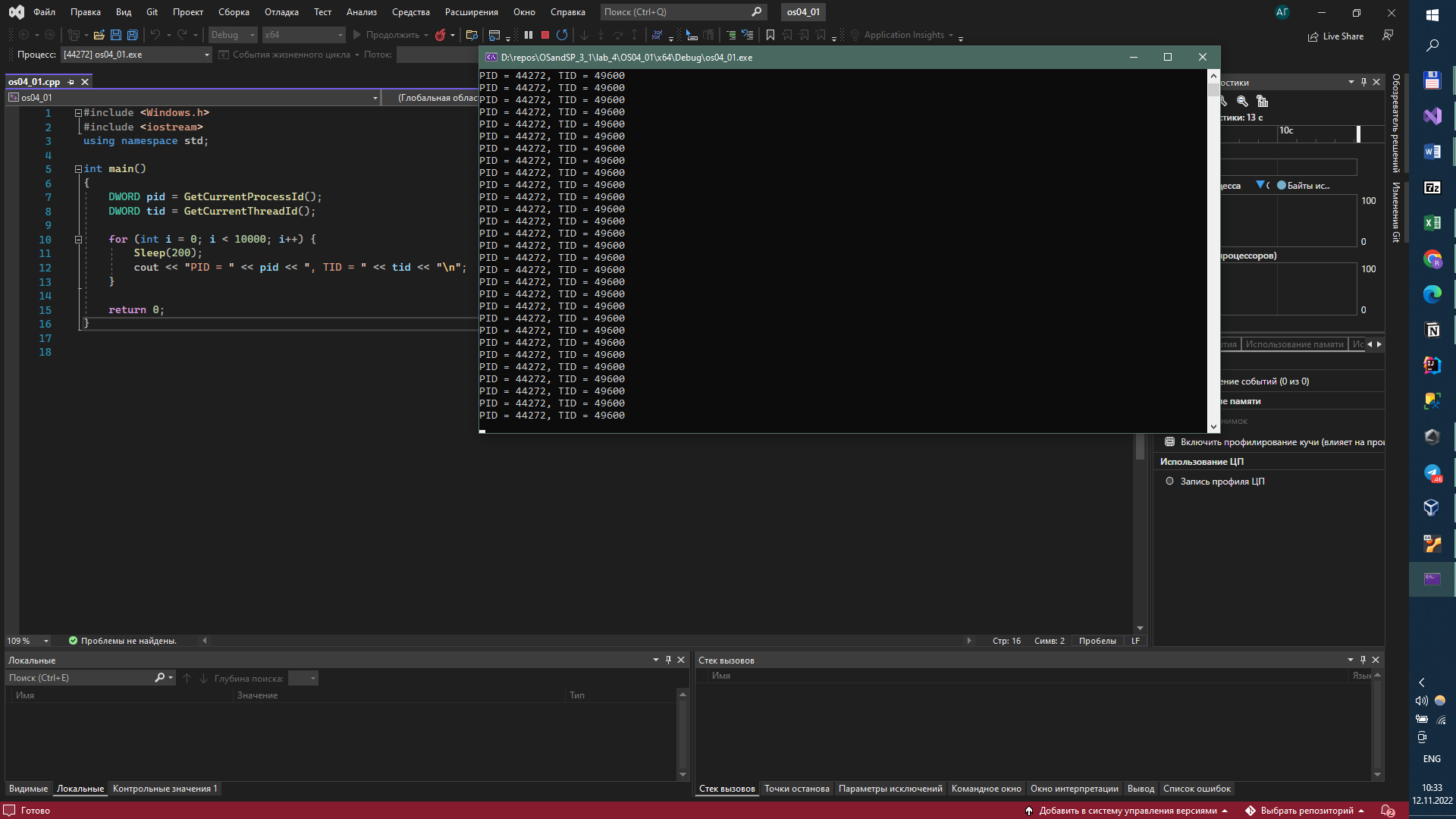
Герман Александр, ФИТ-4

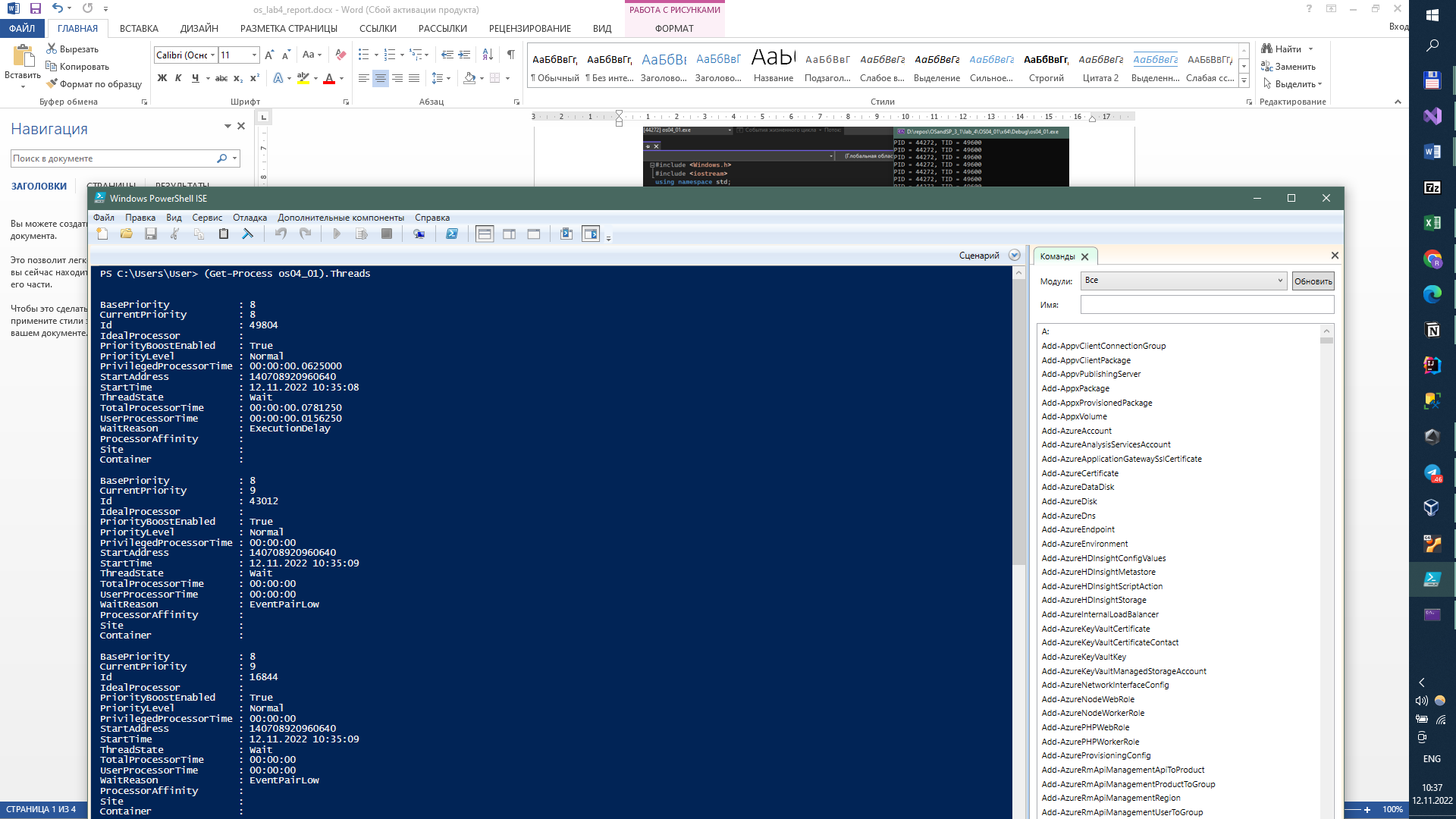
**Лабораторная работа №4**

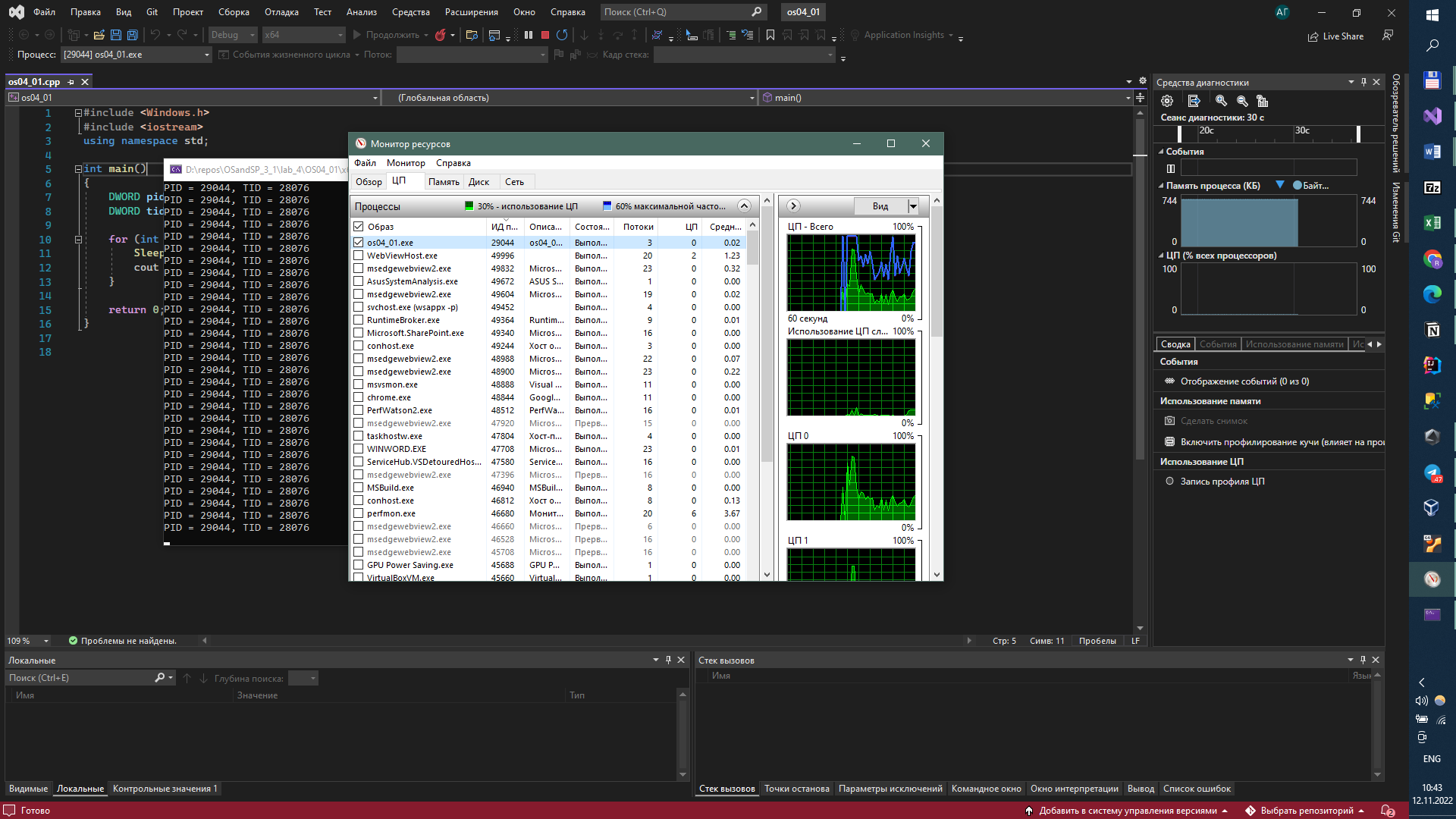
«Потоки»

**Задание 01**

1. Разработайте консольное Windows-приложение **OS04\_01** на языке С++, выполняющее длинный цикл с временной задержкой и с выводом на консоль идентификаторов текущего процесса и текущего потока.
2. Продемонстрируйте информацию об потоках процесса **OS04\_01** с помощью утилит **PowerShell ISE** и **Performance Monitor**.



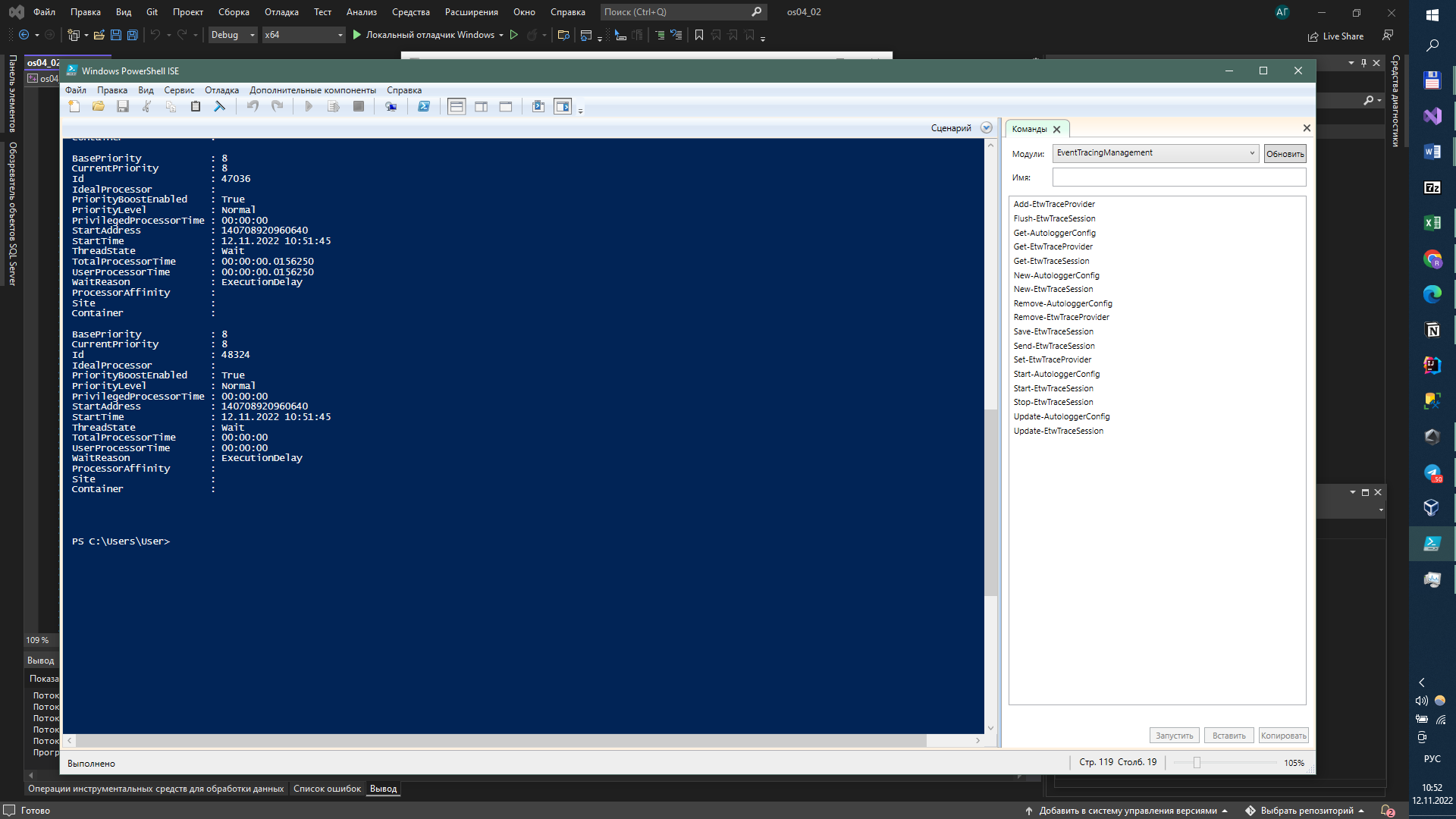
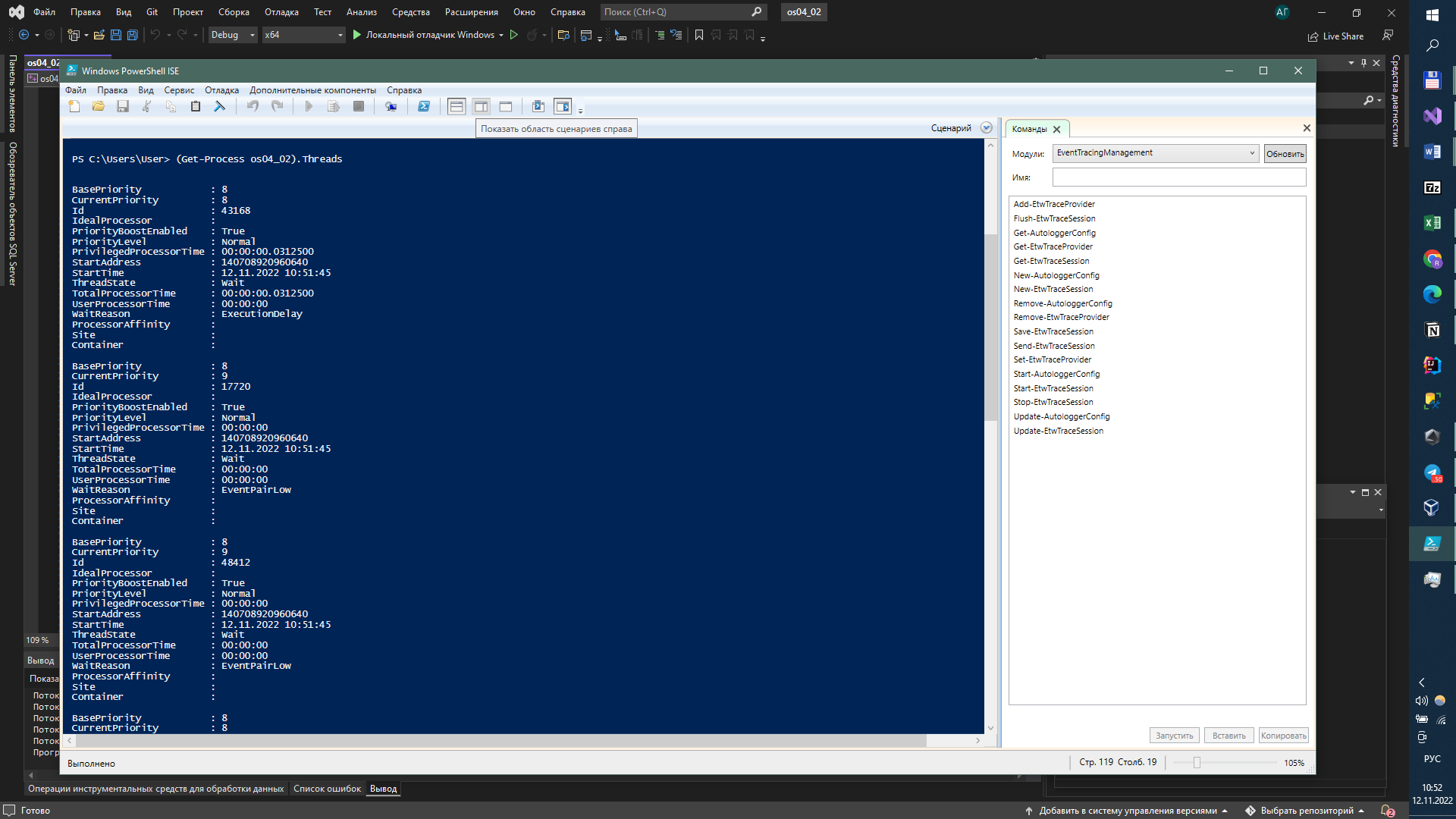




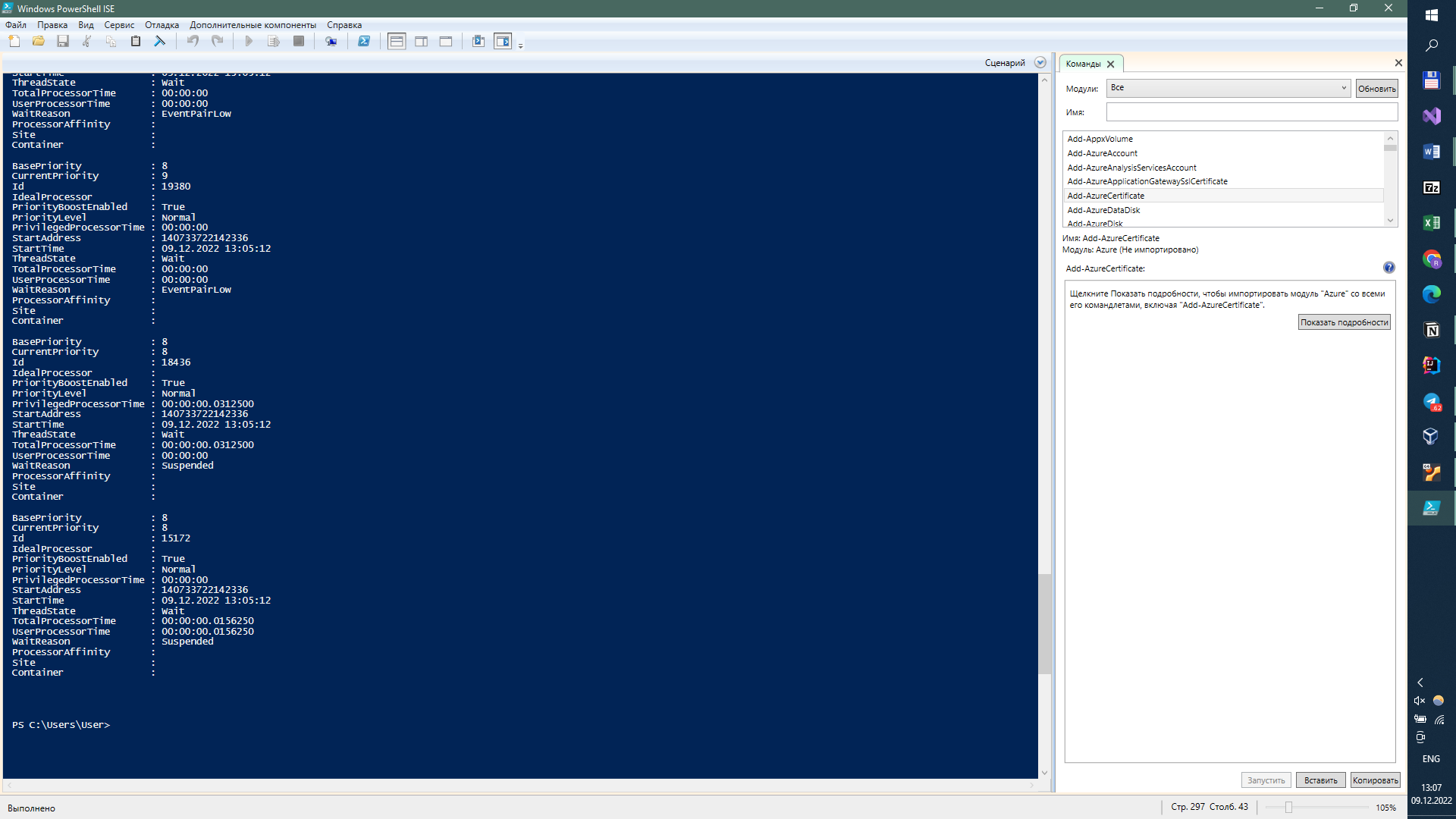
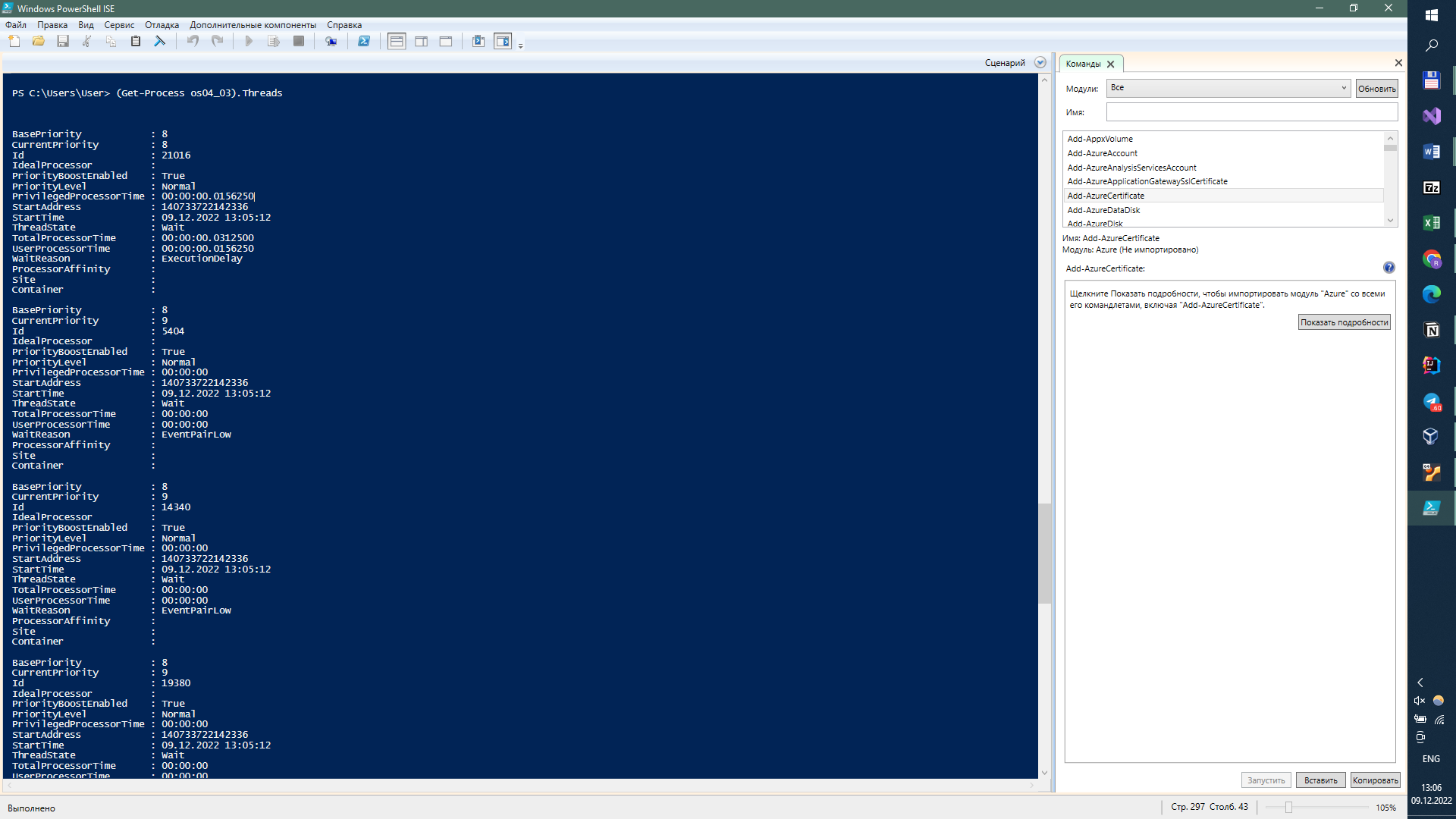
**Задание 02**

1. Разработайте на языке консольное Windows-приложение **OS04\_02** на языке С++, выполняющее цикл 100 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса.
2. Процесс **OS04\_02** должен создать два потока: потоковые функции **OS04\_02\_T1, OS04\_02\_T2.**
3. Поток **OS04\_02\_T1** - выполняет цикл 50 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса и потока.
4. Поток **OS04\_02\_T2** - выполняет цикл 125 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса и потока.
5. Продемонстрируйте информацию об потоках процесса **OS04\_02** с помощью утилит **PowerShell ISE** и **Performance Monitor**.

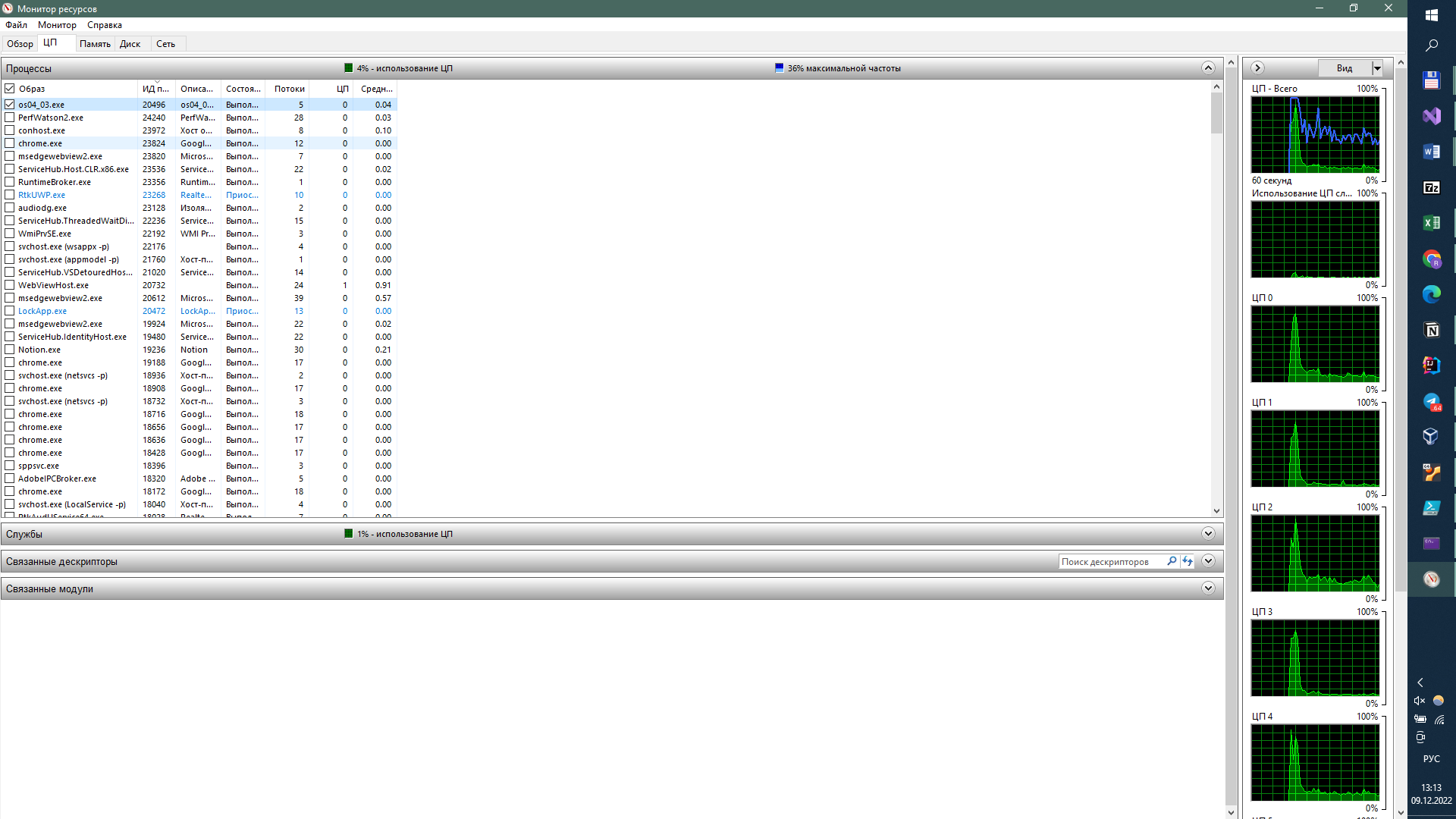
|  |
| --- |
| #include <Windows.h>  #include <iostream>  DWORD os04\_02\_T1() {  DWORD pid = GetCurrentProcessId();  DWORD tid = GetCurrentThreadId();  for (int i = 0; i < 50; ++i) {  Sleep(300);  std::cout << i << ". t1 PID = " << pid << ", TID = " << tid << std::endl;  }  return 0;  }  DWORD os04\_02\_T2() {  DWORD pid = GetCurrentProcessId();  DWORD tid = GetCurrentThreadId();  for (int i = 0; i < 125; ++i) {  Sleep(300);  std::cout << i << ". t2 PID = " << pid << ", TID = " << tid << std::endl;  }  return 0;  }  int main() {  DWORD pid = GetCurrentProcessId();  DWORD tid = GetCurrentThreadId();  DWORD childId\_T1, childId\_T2 = NULL;  HANDLE hChild1 = CreateThread(NULL, 0, (LPTHREAD\_START\_ROUTINE)os04\_02\_T1, NULL, 0, &childId\_T1);  HANDLE hChild2 = CreateThread(NULL, 0, (LPTHREAD\_START\_ROUTINE)os04\_02\_T2, NULL, 0, &childId\_T2);  for (int i = 0; i < 100; ++i) {  Sleep(300);  std::cout << i << ". Parent Thread PID = " << pid << ", TID = " << tid << std::endl;  }    WaitForSingleObject(hChild1, INFINITE);  WaitForSingleObject(hChild2, INFINITE);  CloseHandle(hChild1);  CloseHandle(hChild2);  return 0;  } |



**Задание 03.**

1. Разработайте на языке консольное Windows-приложение **OS04\_03** на языке С++, выполняющее цикл 100 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса.
2. Процесс **OS04\_03** должен создать два потока: потоковые функции **OS04\_03\_T1, OS04\_03\_T2.**
3. Поток **OS04\_03\_T1** - выполняет цикл 50 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса и потока.
4. Поток **OS04\_03\_T2** - выполняет цикл 125 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса и потока.
5. Поток **main** приостанавливает работу потока **OS04\_03\_T1** на20й своей итерации и возобновляют на 60й своей итерации.
6. Поток **main** приостанавливает работу потока **OS04\_03\_T2** на40й своей итерации и возобновляют после окончания собственных итераций (по окончании собственного цикла).
7. Продемонстрируйте информацию об потоках процесса **OS04\_03** с помощью утилит  **PowerShell ISE** и **Performance Monitor**. 

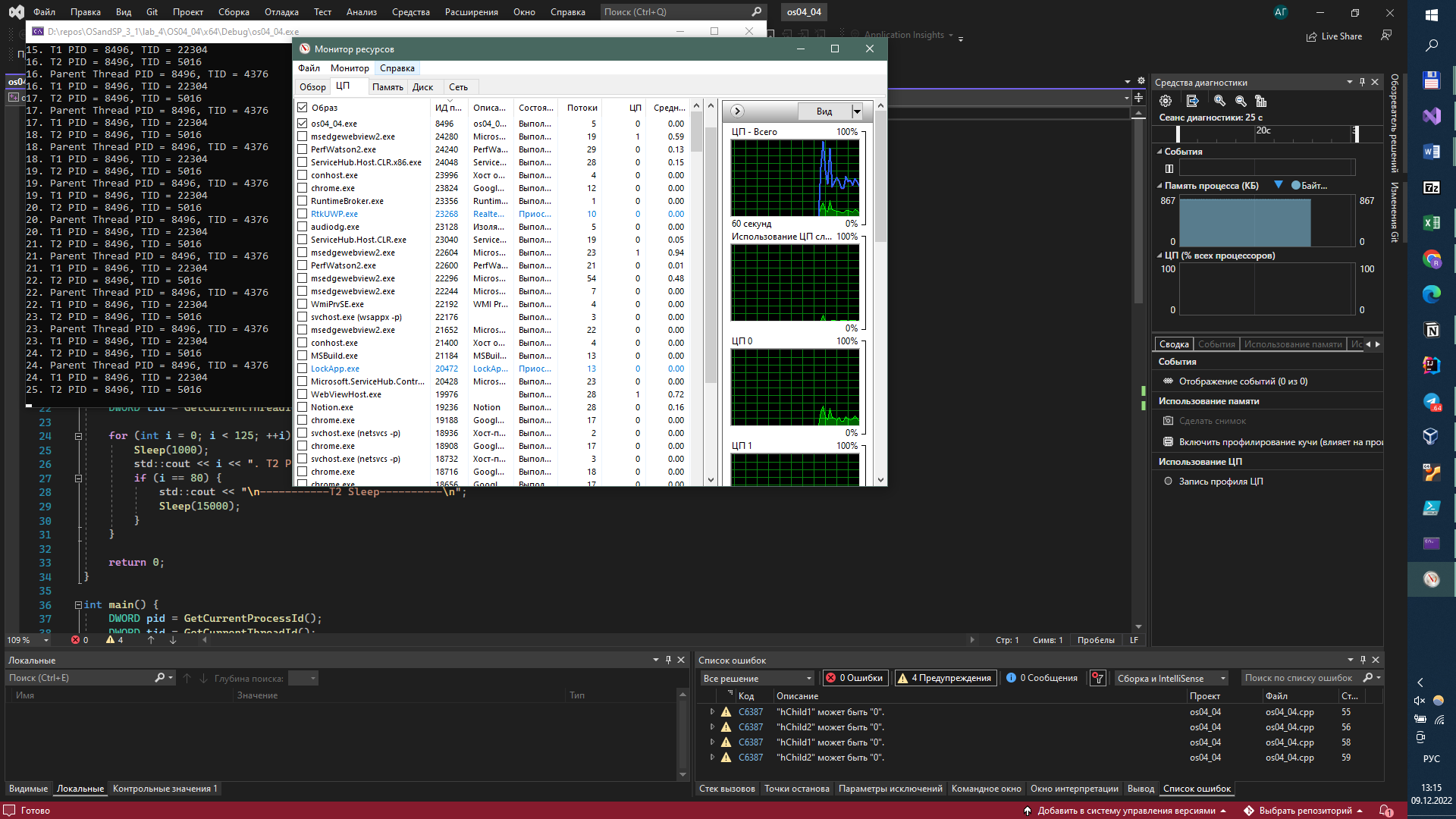
|  |
| --- |
| #include <Windows.h>  #include <iostream>  DWORD os04\_03\_T1() {  DWORD pid = GetCurrentProcessId();  DWORD tid = GetCurrentThreadId();  for (int i = 0; i < 50; ++i) {  Sleep(300);  std::cout << i << ". t1 PID = " << pid << ", TID = " << tid << std::endl;  }  return 0;  }  DWORD os04\_03\_T2() {  DWORD pid = GetCurrentProcessId();  DWORD tid = GetCurrentThreadId();  for (int i = 0; i < 125; ++i) {  Sleep(300);  std::cout << i << ". t2 PID = " << pid << ", TID = " << tid << std::endl;  }  return 0;  }  int main() {  DWORD pid = GetCurrentProcessId();  DWORD tid = GetCurrentThreadId();  DWORD childId\_T1, childId\_T2 = NULL;  HANDLE hChild1 = CreateThread(NULL, 0, (LPTHREAD\_START\_ROUTINE)os04\_03\_T1, NULL, 0, &childId\_T1);  HANDLE hChild2 = CreateThread(NULL, 0, (LPTHREAD\_START\_ROUTINE)os04\_03\_T2, NULL, 0, &childId\_T2);  for (int i = 0; i < 100; ++i) {  Sleep(300);  std::cout << i << ". Parent Thread PID = " << pid << ", TID = " << tid << std::endl;  if (i == 20) {  SuspendThread(hChild1);  SuspendThread(hChild1);  std::cout << "\n-----------Suspend Thread Child1-------------\n";  }  if (i == 60) {  ResumeThread(hChild1);  std::cout << "\n-----------Resume Thread Child1-------------\n";  }  if (i == 40) {  SuspendThread(hChild2);  std::cout << "\n-----------Suspend Thread Child2-------------\n";  }  }  ResumeThread(hChild2);  std::cout << "\n-----------Resume Thread Child2-------------\n";  WaitForSingleObject(hChild1, INFINITE);  WaitForSingleObject(hChild2, INFINITE);  CloseHandle(hChild1);  CloseHandle(hChild2);  return 0;  } |

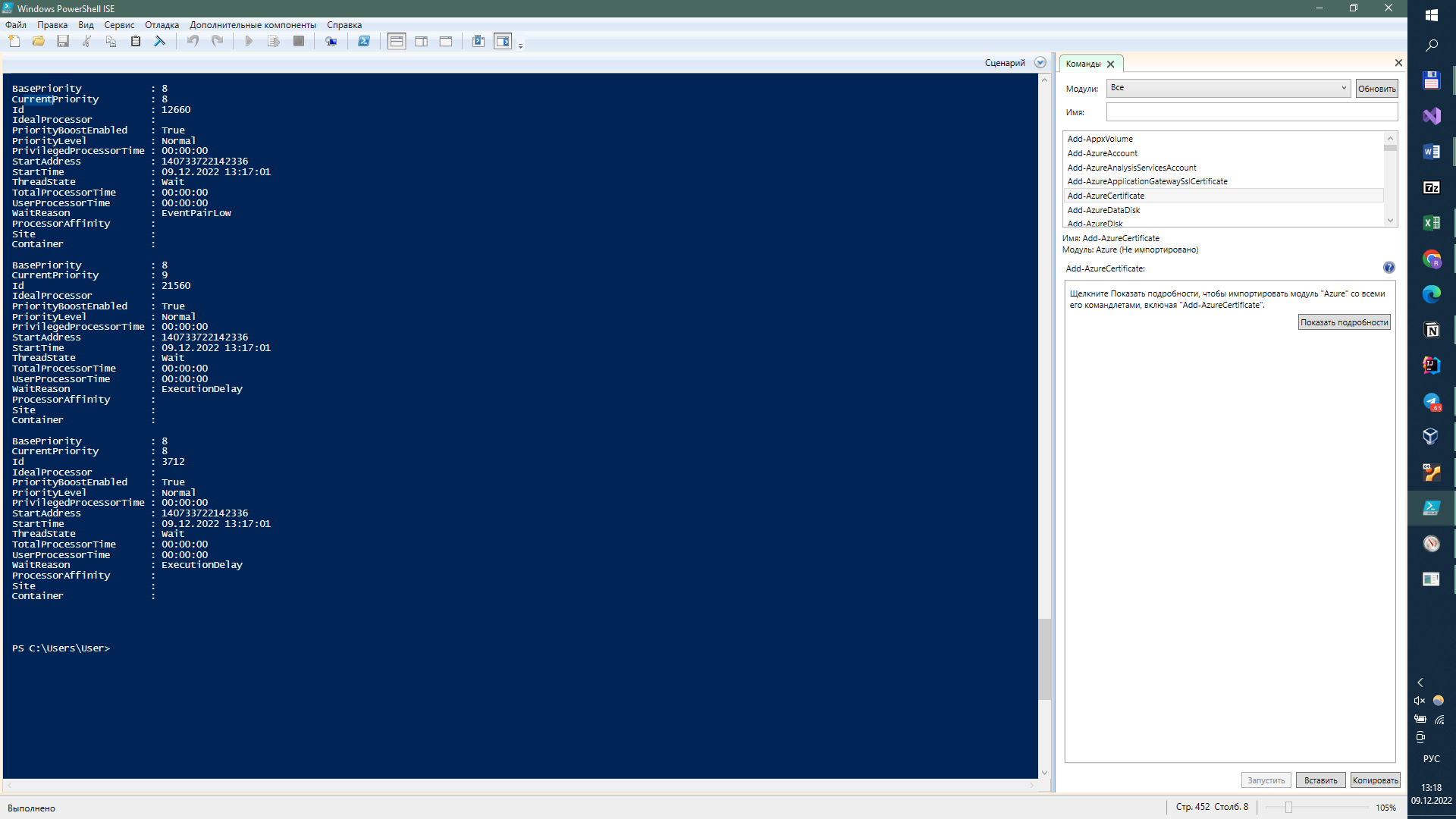
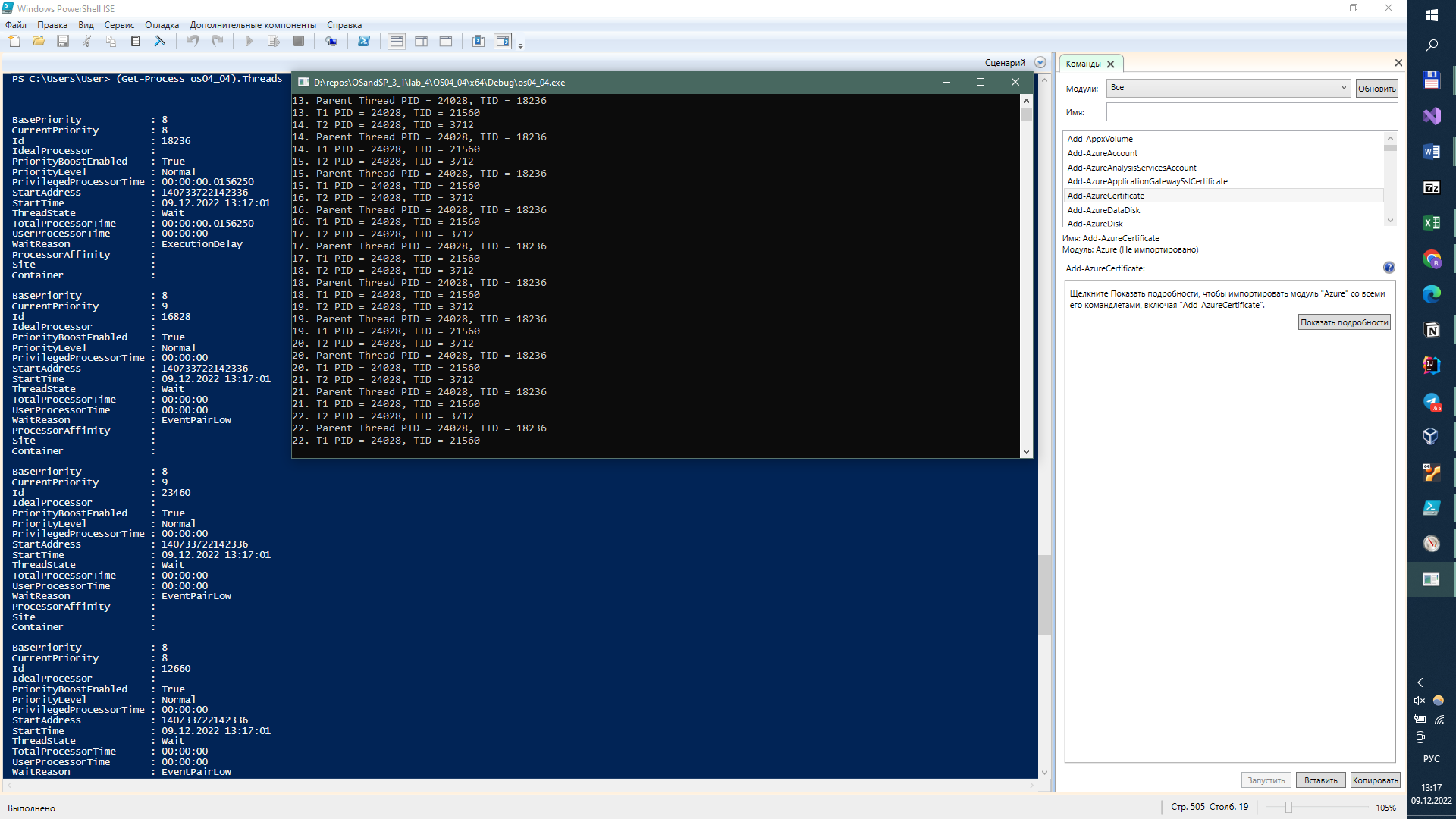


**Задание 04**

1. Разработайте на языке консольное Windows-приложение **OS04\_04** на языке С++, выполняющее цикл 100 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса.
2. Процесс **OS04\_04** должен создать два потока: потоковые функции **OS04\_04\_T1, OS04\_04\_T2.**
3. 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.
4. 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.
5. Поток **main** засыпает на 30й итерации на 10 сек.
6. Продемонстрируйте информацию об потоках процесса **OS04\_04** с помощью утилит  **PowerShell ISE** и **Performance Monitor**.

|  |
| --- |
| #include <Windows.h>  #include <iostream>  DWORD OS04\_04\_T1() {  DWORD pid = GetCurrentProcessId();  DWORD tid = GetCurrentThreadId();  for (int i = 0; i < 50; ++i) {  Sleep(1000);  std::cout << i << ". T1 PID = " << pid << ", TID = " << tid << std::endl;  if (i == 25) {  std::cout << "\n-----------T1 Sleep----------\n";  Sleep(10000);  }  }  return 0;  }  DWORD OS04\_04\_T2() {  DWORD pid = GetCurrentProcessId();  DWORD tid = GetCurrentThreadId();  for (int i = 0; i < 125; ++i) {  Sleep(1000);  std::cout << i << ". T2 PID = " << pid << ", TID = " << tid << std::endl;  if (i == 80) {  std::cout << "\n-----------T2 Sleep----------\n";  Sleep(15000);  }  }  return 0;  }  int main() {  DWORD pid = GetCurrentProcessId();  DWORD tid = GetCurrentThreadId();  DWORD childId\_T1, childId\_T2 = NULL;  HANDLE hChild1 = CreateThread(NULL, 0, (LPTHREAD\_START\_ROUTINE)OS04\_04\_T1, NULL, 0, &childId\_T1);  HANDLE hChild2 = CreateThread(NULL, 0, (LPTHREAD\_START\_ROUTINE)OS04\_04\_T2, NULL, 0, &childId\_T2);  for (int i = 0; i < 100; ++i) {  Sleep(1000);  std::cout << i << ". Parent Thread PID = " << pid << ", TID = " << tid << std::endl;  if (i == 30) {  std::cout << "\n-----------Parent Thread Sleep----------\n";  Sleep(10000);  }  }  WaitForSingleObject(hChild1, INFINITE);  WaitForSingleObject(hChild2, INFINITE);  CloseHandle(hChild1);  CloseHandle(hChild2);  return 0;  } |

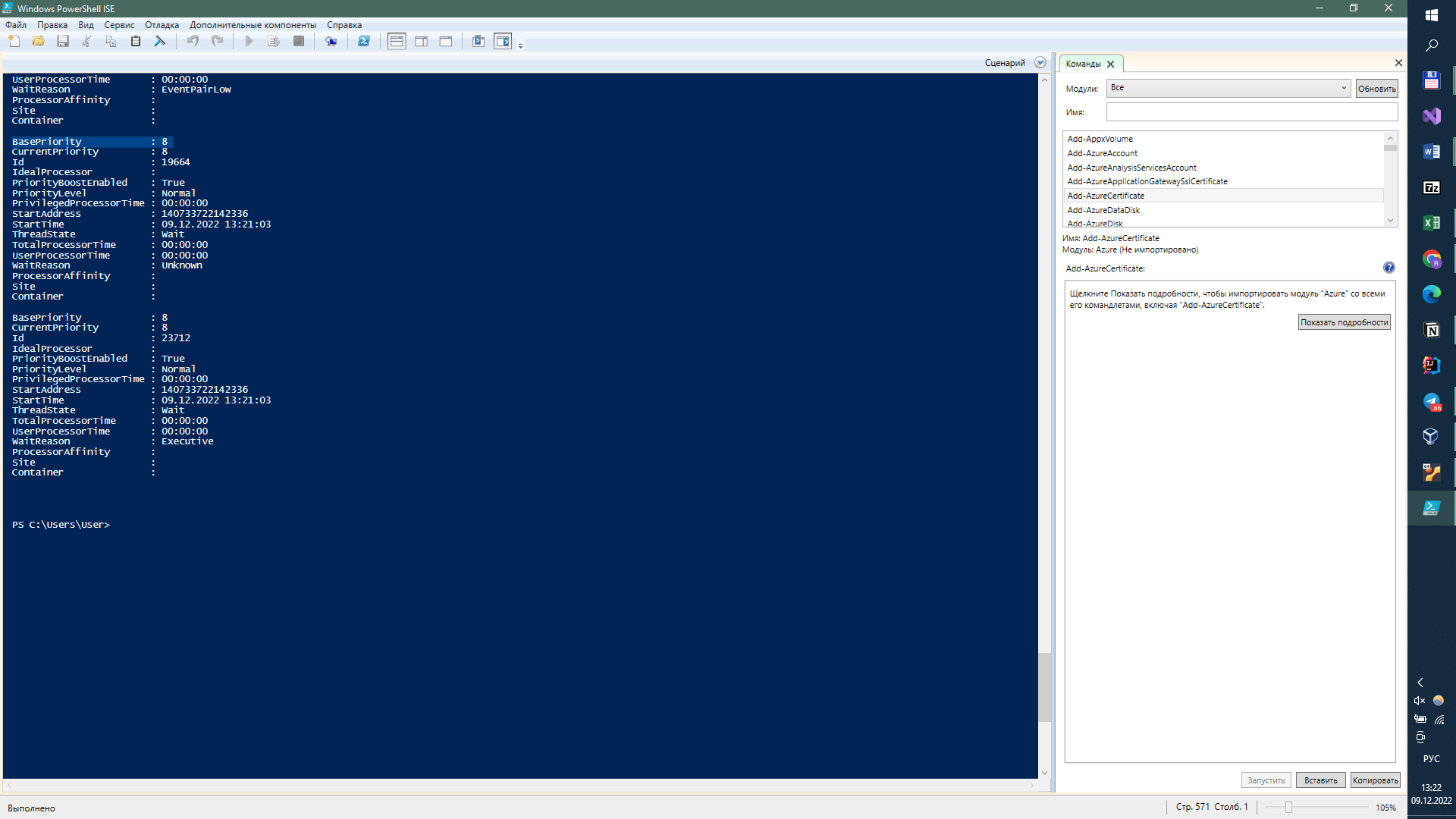
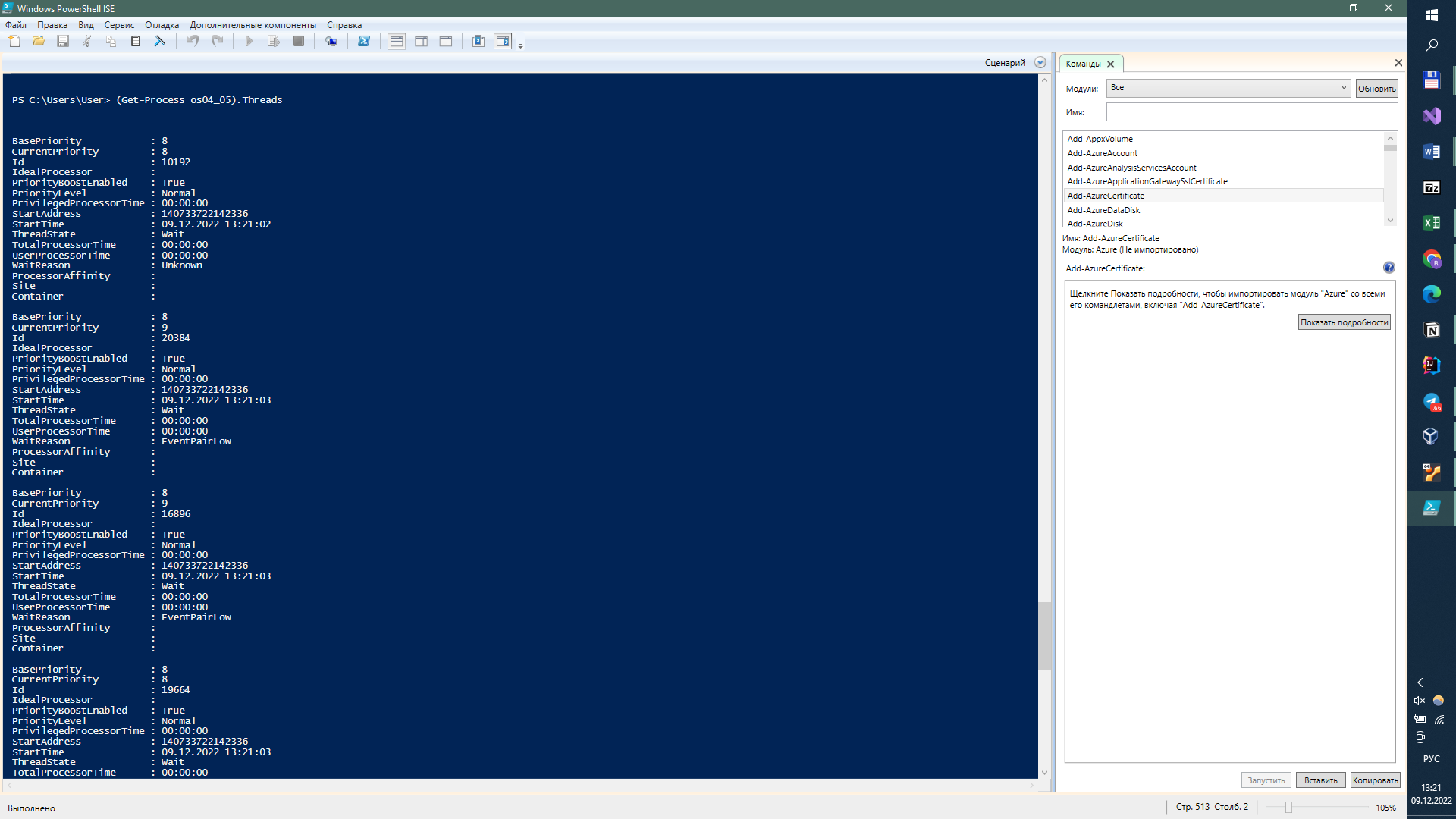




**Задание 05**

1. Разработайте на языке консольное Windows-приложение **OS04\_05** на языке С++, выполняющее цикл 100 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса.
2. Процесс **OS04\_05** должен создать два потока: потоковые функции **OS04\_05\_T1, OS04\_05\_T2.**
3. Поток **OS04\_05\_T1** - выполняет цикл 50 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса и потока.
4. Поток **OS04\_05\_T2** - выполняет цикл 125 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса и потока.
5. Поток **main** завершаетпоток **OS04\_05\_T2** на 40й собственной итерации.
6. Продемонстрируйте информацию об потоках процесса **OS04\_05** с помощью утилит  **PowerShell ISE** и **Performance Monitor**.

|  |
| --- |
| #include <Windows.h>  #include <iostream>  DWORD os04\_05\_T1() {  DWORD pid = GetCurrentProcessId();  DWORD tid = GetCurrentThreadId();  for (int i = 0; i < 50; ++i) {  Sleep(1000);  std::cout << i << ". t1 PID = " << pid << ", TID = " << tid << std::endl;  }  return 0;  }  DWORD os04\_05\_T2() {  DWORD pid = GetCurrentProcessId();  DWORD tid = GetCurrentThreadId();  for (int i = 0; i < 125; ++i) {  Sleep(1000);  std::cout << i << ". t2 PID = " << pid << ", TID = " << tid << std::endl;  }  return 0;  }  int main() {  DWORD pid = GetCurrentProcessId();  DWORD tid = GetCurrentThreadId();  DWORD childId\_T1, childId\_T2 = NULL;  HANDLE hChild1 = CreateThread(NULL, 0, (LPTHREAD\_START\_ROUTINE)os04\_05\_T1, NULL, 0, &childId\_T1);  HANDLE hChild2 = CreateThread(NULL, 0, (LPTHREAD\_START\_ROUTINE)os04\_05\_T2, NULL, 0, &childId\_T2);  for (int i = 0; i < 100; ++i) {  Sleep(200);  std::cout << i << ". Parent Thread PID = " << pid << ", TID = " << tid << std::endl;  if (i == 40) {  std::cout << "\n-----------os04\_05\_T2 - Killed----------\n";  TerminateThread(hChild2, -1);  }  }  WaitForSingleObject(hChild1, INFINITE);  WaitForSingleObject(hChild2, INFINITE);  CloseHandle(hChild1);  CloseHandle(hChild2);  return 0;  } |



**Задание 06**

1. Разработайте на языке консольное Linux-приложение **OS04\_06** на языке С, выполняющее длинный цикл с временной задержкой и с выводом на консоль идентификатора процесса.
2. Продемонстрируйте информацию о потоках процесса **OS04\_06** с помощью утилиты **ps**.

**Задание 07**

1. Разработайте на языке консольное Linux-приложение **OS04\_07** на языке С, выполняющее цикл 100 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса.
2. Процесс **OS04\_07** должен создать поток: потоковая функция **OS04\_07\_T1.**
3. Поток **OS04\_07\_T1** - выполняет цикл 75 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса.
4. Продемонстрируйте информацию о потоках процесса **OS04\_07** с помощью утилиты **ps**.

**Задание 08**

1. Разработайте на языке консольное Linux-приложение **OS04\_08** на языке С, выполняющее цикл 100 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса.
2. Процесс **OS04\_08** должен создать поток: потоковая функция **OS04\_08\_T1.**
3. 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.
4. Поток **main** должен засыпать на 15 сек. на 30й собственной итерации.
5. Продемонстрируйте информацию о потоках процесса **OS04\_08** с помощью утилиты **ps**.

**Задание 09.ответьте на следующие вопросы**

1. Что такое поток управления OS?
2. С помощью каких системных вызовов создаются потоки в Windows и Linux?
3. Что такое системные и пользовательские потоки?
4. Что такое многопоточность?
5. Чем отличаются приоритетная многопоточность от и кооперативной многопоточности?
6. Что такое диспетчеризация потоков управления OS?
7. Что такое контекст потока и для чего он нужен?
8. Перечислите состояния в которых может быть поток и поясните их назначение.
9. Что такое LWP?
10. Что такое потокобезопасность программного кода?
11. Что такое реентерабельность кода?
12. Что такое Fiber?
13. Дайте развернутое определение потока OS.