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 №4**

Дисциплина: Операционные системы

Тема: Потоки

Выполнила:

студентка 3 курса 5 группы

Вовна Ярослава Руслановна

Минск 2024

**Задание 01**

1. int main()

{

DWORD pid = GetCurrentProcessId();

DWORD tid = GetCurrentThreadId();

for (int i = 0; i < 1000; i++)

{

cout << "PID = " << pid << ", TID = " << tid << endl;

Sleep(2000);

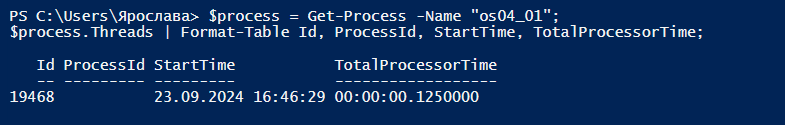
}

}

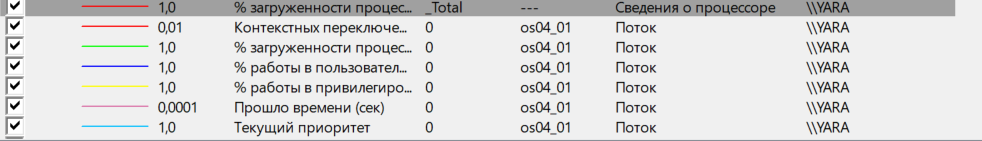
1. powershell ise

$process = Get-Process -Name "os04\_01";

$process.Threads | Format-Table Id, ProcessId, StartTime, TotalProcessorTime;



Performance monitor



**Задание 02**

3-6)

DWORD WINAPI Thread1()

{

DWORD tid = GetCurrentThreadId();

for (short i = 0; i < 50; i++)

{

cout << "PID = " << pid << ", ID of thread1 = " << tid << endl;

Sleep(1000);

}

return 0;

}

DWORD WINAPI Thread2()

{

DWORD tid = GetCurrentThreadId();

for (short i = 0; i < 125; i++)

{

cout << "PID = " << pid << ", ID of thread2 = " << tid << endl;

Sleep(1000);

}

return 0;

}

int main()

{

pid = GetCurrentProcessId();

DWORD os04\_02ID = GetCurrentThreadId();

DWORD os04\_02\_T1ID = NULL;

DWORD os04\_02\_T2ID = NULL;

HANDLE os04\_02\_T1 = CreateThread(NULL, 0, (LPTHREAD\_START\_ROUTINE)Thread1, NULL, 0, &os04\_02\_T1ID);

HANDLE os04\_02\_T2 = CreateThread(NULL, 0, (LPTHREAD\_START\_ROUTINE)Thread2, NULL, 0, &os04\_02\_T2ID);

for (short i = 0; i < 100; ++i)

{

cout << "PID = " << pid << ", Thread main ID = " << os04\_02ID << endl;

Sleep(1000);

}

WaitForSingleObject(os04\_02\_T1, INFINITE);

WaitForSingleObject(os04\_02\_T2, INFINITE);

CloseHandle(os04\_02\_T1);

CloseHandle(os04\_02\_T2);

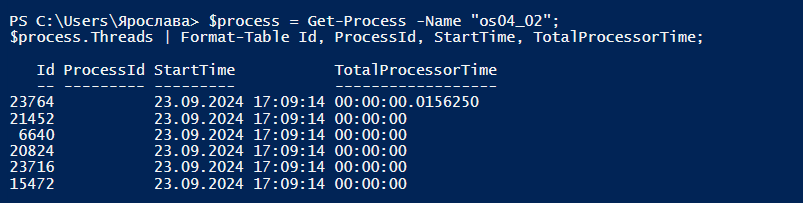
system("pause");

return 0;

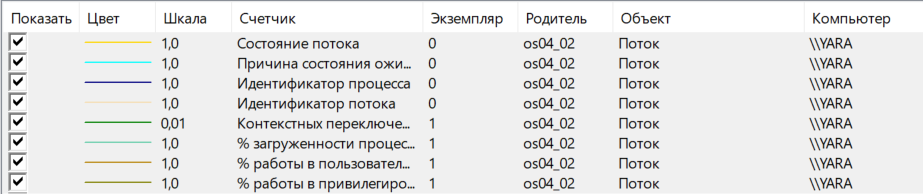
}

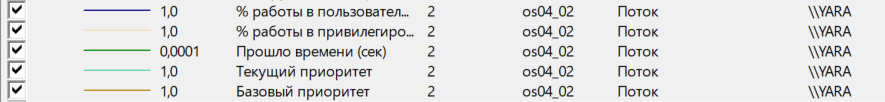


1. powershell ise



Performance monitor





**Задание 03**

8-13)

DWORD WINAPI Thread1()

{

DWORD tid = GetCurrentThreadId();

for (short i = 0; i < 50; ++i)

{

cout << "PID = " << pid << ", ID of thread1 = " << tid << endl;

Sleep(1000);

}

return 0;

}

DWORD WINAPI Thread2()

{

DWORD tid = GetCurrentThreadId();

for (short i = 0; i < 125; ++i)

{

cout << "PID = " << pid << ", ID of thread2 = " << tid << endl;

Sleep(1000);

}

return 0;

}

int main()

{

pid = GetCurrentProcessId();

DWORD mainID = GetCurrentThreadId();

DWORD os04\_03\_T1ID = NULL;

DWORD os04\_03\_T2ID = NULL;

HANDLE os04\_03\_T1 = CreateThread(NULL, 0, (LPTHREAD\_START\_ROUTINE)Thread1, NULL, 0, &os04\_03\_T1ID);

HANDLE os04\_03\_T2 = CreateThread(NULL, 0, (LPTHREAD\_START\_ROUTINE)Thread2, NULL, 0, &os04\_03\_T2ID);

for (short i = 1; i <= 100; ++i)

{

switch (i)

{

case 20: SuspendThread(os04\_03\_T1); break;

case 60: ResumeThread(os04\_03\_T1); break;

case 40: SuspendThread(os04\_03\_T2); break;

case 100: ResumeThread(os04\_03\_T2); break;

}

cout << i << " - PID = " << pid << ", Thread main ID = " << mainID << endl;

Sleep(1000);

}

WaitForSingleObject(os04\_03\_T1, INFINITE);

WaitForSingleObject(os04\_03\_T1, INFINITE);

CloseHandle(os04\_03\_T1);

CloseHandle(os04\_03\_T1);

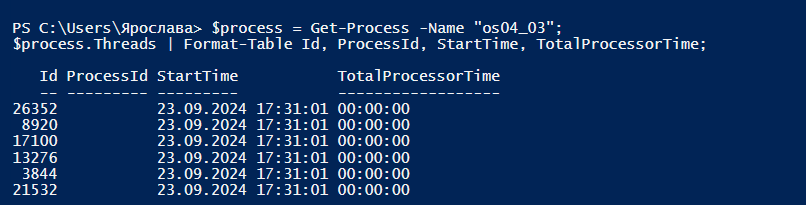
system("pause");

return 0;

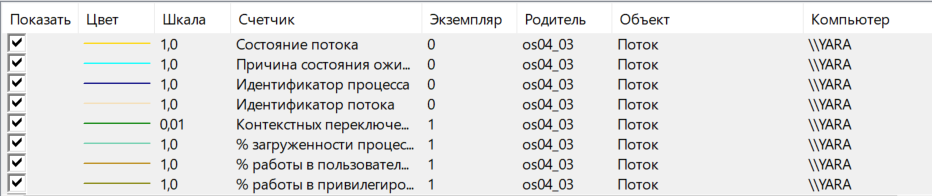
}

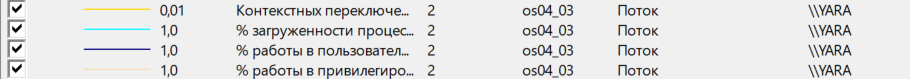


14) powershell ise



Performance monitor





**Задание 04**

15-19)

DWORD WINAPI Thread1()

{

DWORD tid = GetCurrentThreadId();

for (short i = 0; i < 50; ++i)

{

if (i == 25)

{

Sleep(10000);

}

cout << "PID = " << pid << ", ID of thread1 = " << tid << endl;

Sleep(1000);

}

return 0;

}

DWORD WINAPI Thread2()

{

DWORD tid = GetCurrentThreadId();

for (short i = 0; i < 125; ++i)

{

if (i == 80)

{

Sleep(15000);

}

cout << "PID = " << pid << ", ID of thread2 = " << tid << endl;

Sleep(1000);

}

return 0;

}

int main()

{

pid = GetCurrentProcessId();

DWORD mainID = GetCurrentThreadId();

DWORD os04\_04\_T1ID = NULL;

DWORD os04\_04\_T2ID = NULL;

HANDLE os04\_04\_T1 = CreateThread(NULL, 0, (LPTHREAD\_START\_ROUTINE)Thread1, NULL, 0, &os04\_04\_T1ID);

HANDLE os04\_04\_T2 = CreateThread(NULL, 0, (LPTHREAD\_START\_ROUTINE)Thread2, NULL, 0, &os04\_04\_T2ID);

for (short i = 1; i <= 100; ++i)

{

if (i == 30)

{

Sleep(10000);

}

cout << i << " - PID = " << pid << ", Thread main ID = " << mainID << endl;

Sleep(1000);

}

WaitForSingleObject(os04\_04\_T1, INFINITE);

WaitForSingleObject(os04\_04\_T2, INFINITE);

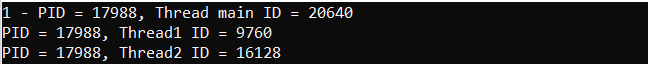
CloseHandle(os04\_04\_T1);

CloseHandle(os04\_04\_T2);

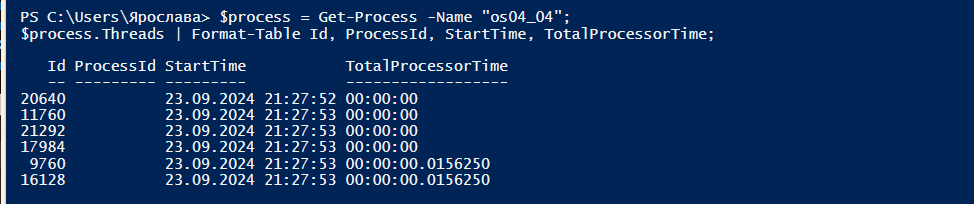
system("pause");

return 0;

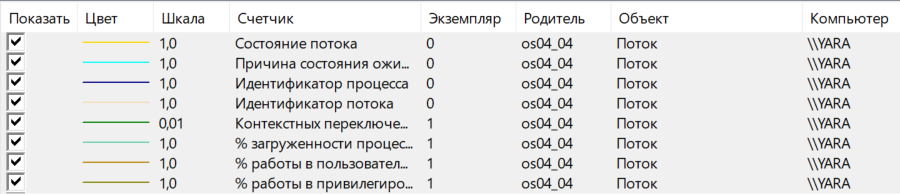
}

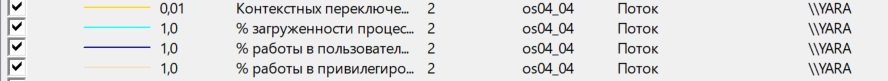


20) powershell ise



Performance monitor





**Задание 05**

21-25)

DWORD WINAPI Thread1()

{

DWORD tid = GetCurrentThreadId();

for (short i = 0; i < 50; ++i)

{

cout << "PID = " << pid << ", ID of thread1 = " << tid << endl;

Sleep(1000);

}

return 0;

}

DWORD WINAPI Thread2()

{

DWORD tid = GetCurrentThreadId();

for (short i = 0; i < 125; ++i)

{

cout << "PID = " << pid << ", ID of thread2 = " << tid << endl;

Sleep(1000);

}

return 0;

}

int main()

{

pid = GetCurrentProcessId();

DWORD mainID = GetCurrentThreadId();

DWORD os04\_05\_T1ID = NULL;

DWORD os04\_05\_T2ID = NULL;

HANDLE os04\_05\_T1 = CreateThread(NULL, 0, (LPTHREAD\_START\_ROUTINE)Thread1, NULL, 0, &os04\_05\_T1ID);

HANDLE os04\_05\_T2 = CreateThread(NULL, 0, (LPTHREAD\_START\_ROUTINE)Thread2, NULL, 0, &os04\_05\_T2ID);

for (short i = 1; i <= 100; ++i)

{

if (i == 40)

{

TerminateThread(os04\_05\_T2, 0);

}

cout << i << " - PID = " << pid << ", Thread main ID = " << mainID << endl;

Sleep(1000);

}

WaitForSingleObject(os04\_05\_T1, INFINITE);

WaitForSingleObject(os04\_05\_T2, INFINITE);

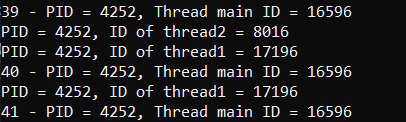
CloseHandle(os04\_05\_T1);

CloseHandle(os04\_05\_T2);

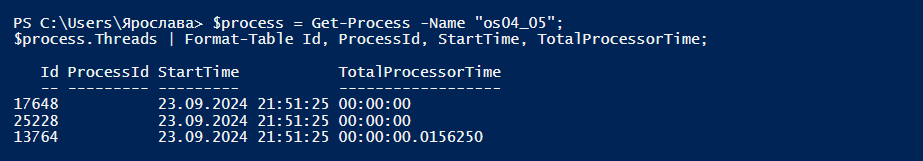
system("pause");

return 0;

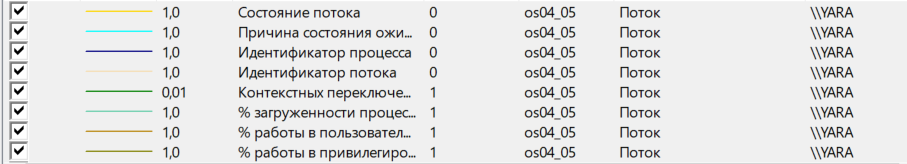
}

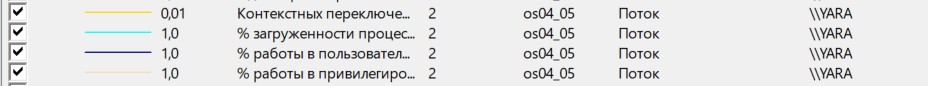


26) powershell ise



Performance monitor





**Задание 06**

27)

int main()

{

pid\_t pid = getpid();

pid\_t tid = gettid();

for (short i = 1; i <= 10000; ++i)

{

printf("%d. PID = %d, TID = %d\n", i, pid, tid);

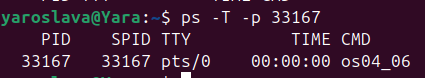
sleep(1);

}

exit(0);

}

28)



**Задание 07**

29-31)

void\* Thread(void\* arg)

{

pid\_t pid = getpid();

pid\_t tid = syscall(SYS\_gettid);

for (int i = 1; i <= 75; ++i)

{

printf("%d. PID = %d, ID of thread = %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, Thread, NULL);

for (int i = 1; i <= 100; ++i)

{

printf("%d. PID = %d, Thread main ID = %d\n", i, pid, tid);

sleep(1);

}

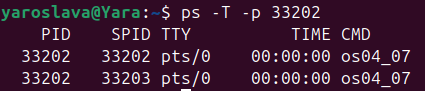
int status = pthread\_join(a\_thread, (void\*\*)&thread\_result);

exit(0);

}

32)

C:\Users\Ярослава\AppData\Local\Microsoft\Windows\INetCache\Content.Word\3.png



**Задание 08**

33-36)

void\* Thread(void\* arg)

{

pid\_t pid = getpid();

pid\_t tid = syscall(SYS\_gettid);

for (int i = 1; i <= 75; ++i)

{

printf("%d. PID = %d, ID of thread = %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, Thread, NULL);

for (int i = 1; i <= 100; ++i)

{

printf("%d. PID = %d, Thread main ID = %d\n", i, pid, tid);

sleep(1);

if (i == 30)

sleep(15);

}

int status = pthread\_join(a\_thread, (void\*\*)&thread\_result);

exit(0);

}

37)

