IE6 OSL – Lab 4

Authors: Thishan Warnakulasooriya, Celestine Machuca.

Supervisor: Prof. Dr.-Ing. Holger Gräßner

[Research Microsoft Developer Network (MSDN) 1](#_Toc139136827)

[Task 1: Synchronization 3](#_Toc139136828)

[Complete the project Control: 3](#_Toc139136829)

[Complete the project Processing: 3](#_Toc139136830)

[Complete the project InputOutput: 4](#_Toc139136831)

[Task 2: Process priorities and processor assignment 4](#_Toc139136832)

[Task Manager Setting Priority to low for Input Output/ HIGH for testfile 6](#_Toc139136833)

[Task Manager Setting Priority to high for Input Output/ LOW for testfile 7](#_Toc139136834)

[References 8](#_Toc139136835)

[[1] 9](#_Toc139136836)

[[2] 9](#_Toc139136837)

[[3] 9](#_Toc139136838)

[[4] 9](#_Toc139136839)

[[5] 9](#_Toc139136840)

# Research Microsoft Developer Network (MSDN)

* How to get the current process handle?

The **GetCurrentProcess()** function is used to retrieve a pseudo handle for the current process. This pseudo handle has the PROCESS\_ALL\_ACCESS access right to the process object, and it does not need to be closed when it is no longer needed [1].

* How to get the current process ID

you can use the GetCurrentProcessId function, which is declared in the processthreadsapi.h header file.

int processid = GetProcessId(hProcess);

• How does the priorities management work?

Each thread is assigned a scheduling priority, and these range from zero (lowest priority) to 31 (highest priority). The system treats all threads with the same priority as equal and assigns time slices in a round-robin fashion to all threads with the highest priority. If none of these threads are ready to run, the system assigns time slices to all threads with the next highest priority. If a higher-priority thread becomes available to run, the system ceases to execute the lower-priority thread and assigns a full time slice to the higher-priority thread. The priority of each thread is determined by the priority class of its process and the priority level of the thread within the priority class of its process [2].

• How to set process priorities?

The priority class of a process can be set using the SetPriorityClass function. The priority level of a thread can be adjusted using the SetThreadPriority function. The priority class and priority level can be used to manage the overall priorities of the threads in a process [3]

• How can I assign a thread to a specific CPU core?

You can set the core affinity by applying a mask as follows:

SetProcessAffinityMask(hProcess, processAffinityMask)

The affinity mask for the threads of the process [3].

A screenshot of a computer program

Description automatically generated with medium confidence

• How can I find out the CPU core assigned to a specific thread?

You can make use of the pid and the windows task manager to search for it.

• What about ProcessAffinityMask and PriorityClass?

A picture containing text, screenshot, font

Description automatically generated

Figure 1 PriorityAffinityMask Microsoft MSDN[4]

A picture containing text, screenshot, font, menu

Description automatically generated

Figure 2 Priority Class Microsoft MSDN [5]

# Task 1: Synchronization

## Complete the project Control:

* Add events and event handling.
* Events should be named uniquely.
* This process should do some output, e. g. to inform the user about a start of the
* calculation.
* This process receives events if a trigger signal has been recognized or if the
* calculation has been finished.

## Complete the project Processing:

* Add events and event handling.
* Events should be named uniquely.
* This process must inform the control process about a completed calculation.
* This process receives a command to start the calculation.

## Complete the project InputOutput:

* The Meilhaus IO card is already included in the start project.
* Create events and event handling.
* Events should be named uniquely.
* This process already includes the Meilhaus driver: IRQ-trigger (Pin48) and digital.
* Output (Pin 29) of the Meilhaus ME-4660 IO card.
* This process must inform the control process about a new trigger. As soon, as a
* trigger is detected at the digital input of the IO card, the interrupt function.
* IrqCallback() will be called. Here you signal an event.
* This process receives a command to do a digital output.

# Task 2: Process priorities and processor assignment

1. Add some code to show the current priority and the assigned processor core of your processes.

2. You will find a computationally intensive program: Studio project testfile.sln. Add some code to get 3 threads, each assigned to a different processor core. Note: You can observe the workload of each processor core by running the windows resource monitor.

a. Start your program via batch file start.bat first (which processor hosts your processes?) and your test program testfile.prg. Observe the program’s behavior.

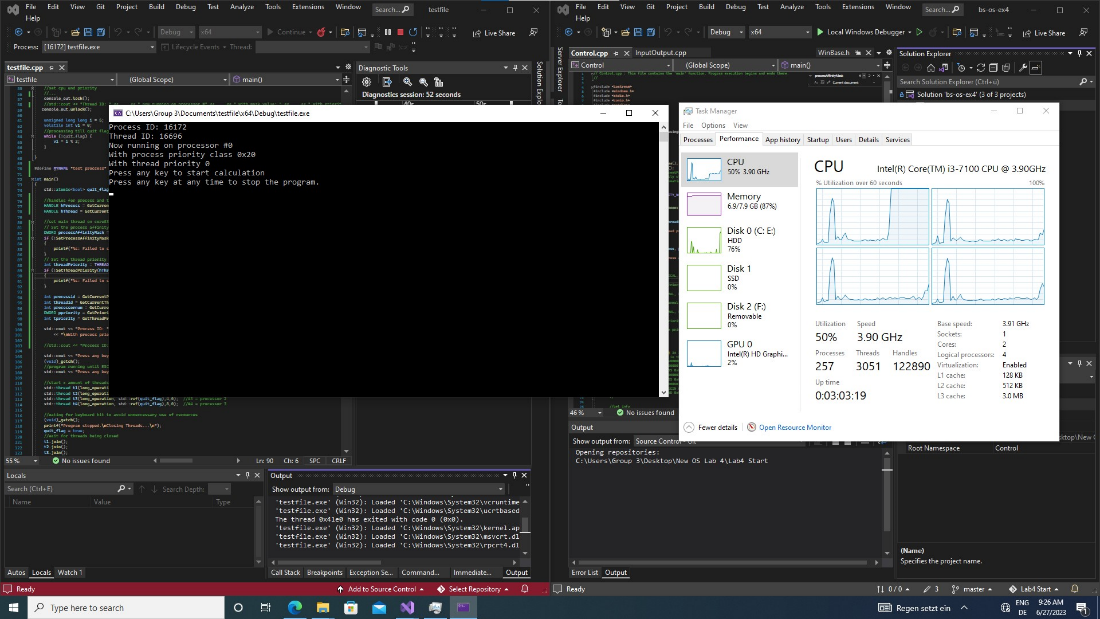


Figure 3 Task Manager View of Logical Core Usage

c. Switch between the 4 console windows by using the mouse. Describe the effect to the signal output.

the output is being affected by the switching between the consoles, we presume that windows is giving the console with the focus more resources than the others as said on the lecture.

3. Repeat the tasks a) b) and c); now with 4 threads in your test program.

a. Start your processes with increased priority (ABOVE\_NORMAL\_PRIORITY\_CLASS) via start.bat first and then start the test program (same priority as above). Observe the program’s behavior. Switch between the 4 console windows by using the mouse. Describe the effect of the signal output.

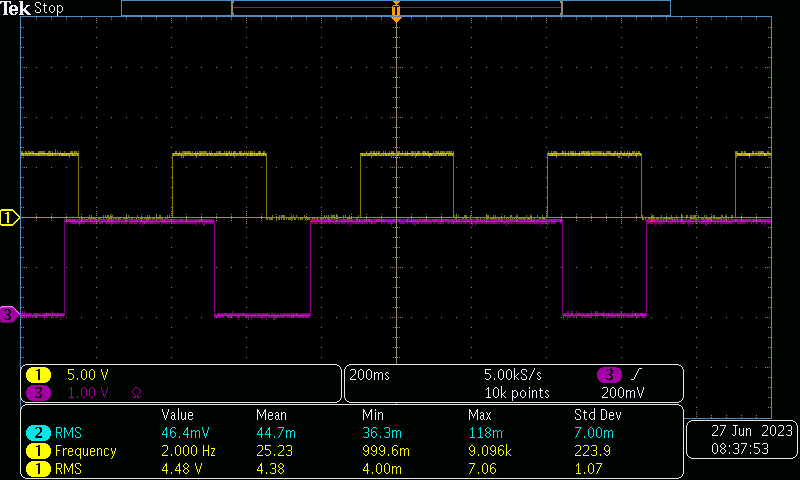


Figure 4 Adobe Normal Priority

b. Start the test program with lower priority (BELOW\_NORMAL\_PRIORITY\_CLASS) first and then your program via start.bat. Observe the program’s behavior. Switch between the 4 console windows by using the mouse. Describe the effect of the signal output.

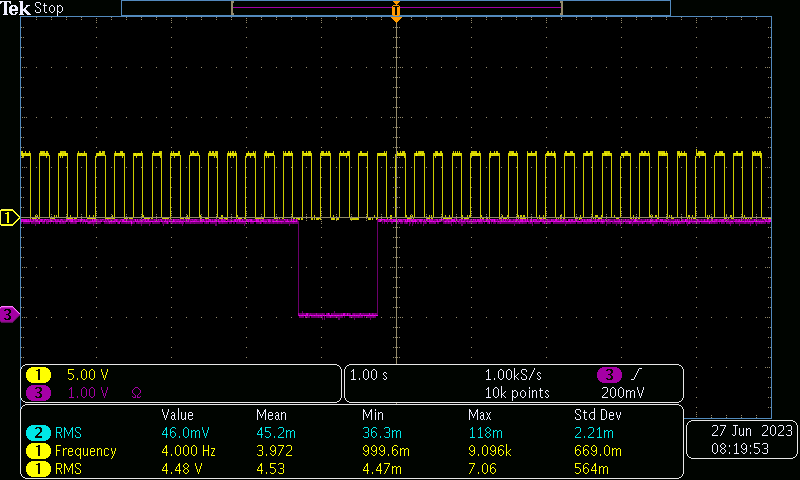


Figure 5 Below Normal Priority

4. You can also change priorities and processor assignment with the Windows task manager. Test different priorities and assignments.

### Task Manager Setting Priority to low for Input Output/ HIGH for testfile

A screenshot of a computer

Description automatically generated

Figure 6 Task Manager Setting Priority to low for Input Output

A screenshot of a computer

Description automatically generatedFigure 7 Task Manager Setting Priority to high for testfile

A screen shot of a computer

Description automatically generated with low confidence

Figure 8 Oscilloscope output

### Task Manager Setting Priority to high for Input Output/ LOW for testfile

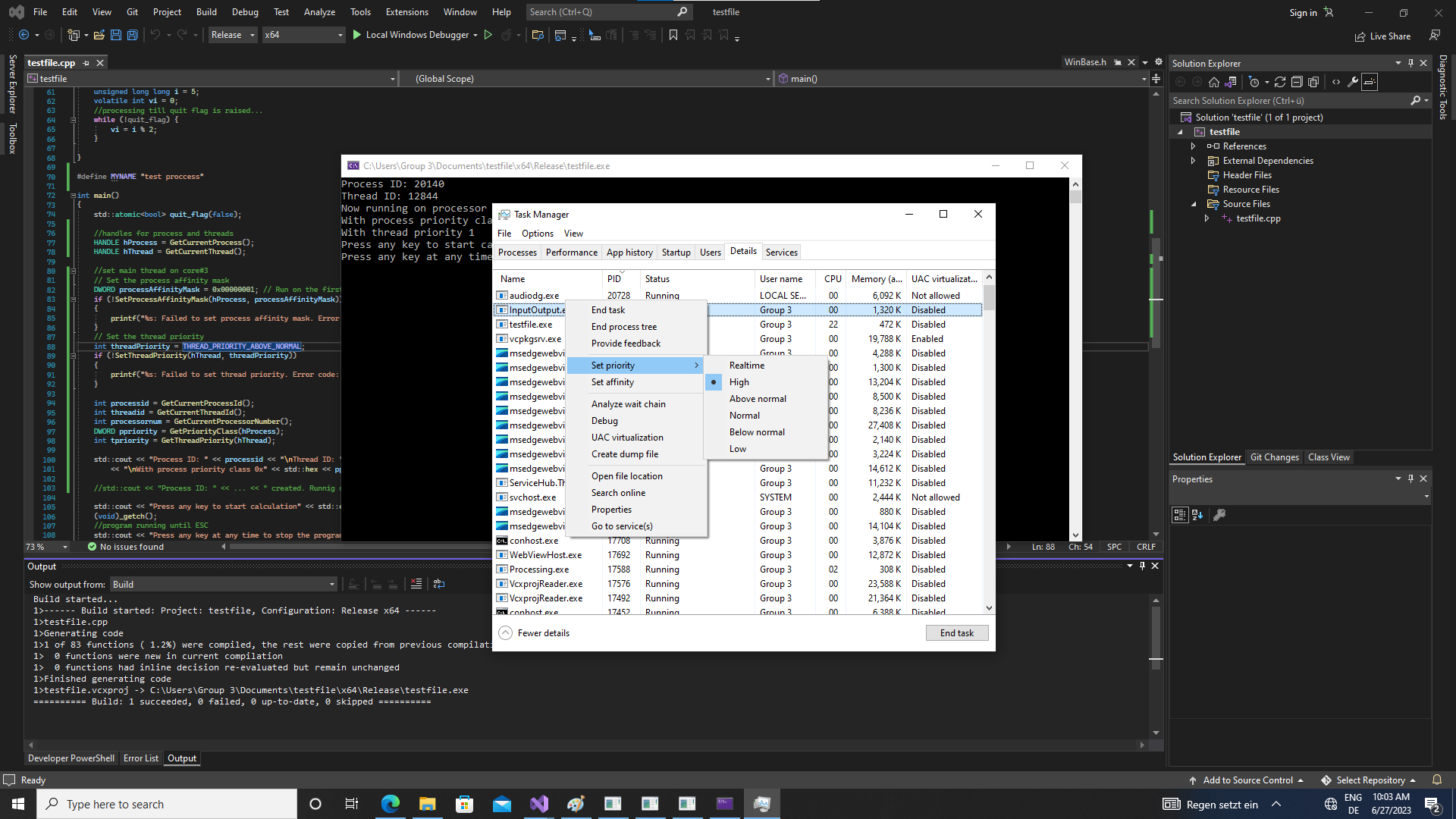


Figure 9 Task Manager Setting Priority to high for Input Output

A screenshot of a computer

Description automatically generated

Figure 10 Testfile set to low windows Taskmanager

A screen shot of a computer

Description automatically generated with low confidence

Figure 11 Oscilloscope output

References

[1] <https://learn.microsoft.com/en-us/windows/win32/api/processthreadsapi/nf-processthreadsapi-getcurrentprocess> GetCurrentProcess

[2] <https://learn.microsoft.com/en-us/windows/win32/api/processthreadsapi/nf-processthreadsapi-getcurrentprocessid> GetCurrentProcessId

[3] <https://learn.microsoft.com/en-us/windows/win32/api/processthreadsapi/nf-processthreadsapi-getcurrentthread> GetCurrentThread

[4] <https://learn.microsoft.com/en-usdotnet/api/system.diagnostics.processthread.processoraffinity?view=net-7.0> processor affinity

[5] <https://learn.microsoft.com/en-us/windows/win32/api/processthreadsapi/nf-processthreadsapi-setpriorityclass> priority class