Учреждения образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

Кафедра информационных систем и технологий

Специальность 1-40 01 01 Программное обеспечение информационной безопасности мобильных систем

**Отчёт по лабораторной работе №5**

Дисциплина: Операционные системы

Тема: Диспетчеризация

Выполнила:

студентка 3 курса 5 группы

Городилина Анастасия Сергеевна

Минск 2024

**Задание 01**

Разработайте консольное Windows-приложение OS05\_01 на языке С++, выводящее на консоль следующую информации:

- идентификатор текущего процесса;

- идентификатор текущего (main) потока;

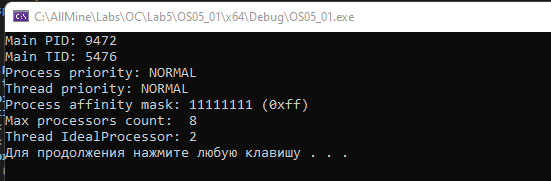
- приоритет (приоритетный класс) текущего процесса;

- приоритет текущего потока;

- маску (affinity mask) доступных процессу процессоров в двоичном виде

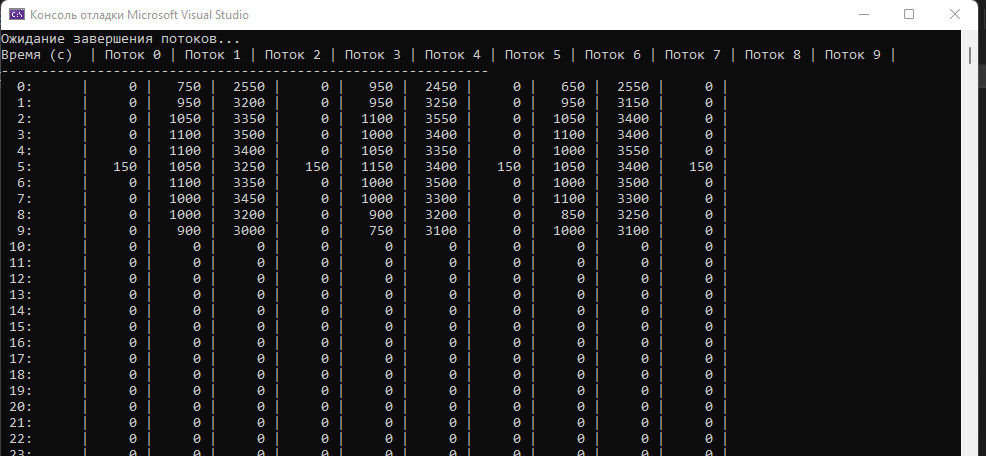
- количество процессоров, доступных процессу;

- процессор, назначенный текущему потоку.



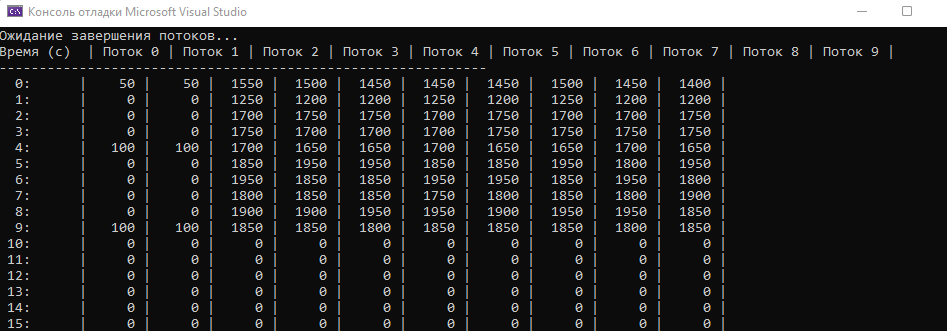
**Задание 02**

Создайте консольное Windows OS05\_02 на языке С#, взяв за основу приложение OS04\_07 из Лабораторной работы №4. Измените метод Main таким образом, чтобы потоки 0, 3, 6 и т.д. запускались с минимальным приоритетом потока, а потоки 2, 5, 8... – с максимальным. Класс приоритета процесса оставьте по умолчанию (Normal).

****

**Задание 03**

Создайте консольное Windows OS05\_03 на языке С#, взяв за основу приложение OS05\_02 из настоящей работы. На этот раз только несколько потоков запустите на наименьшем приоритете потока, а остальные – на наибольшем.



**Задание 04**

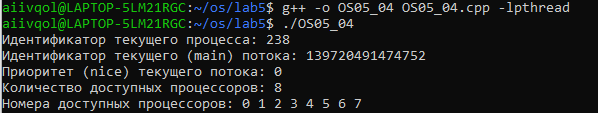
Разработайте консольное Linux-приложение OS05\_04 на

языке С++, выводящее на консоль следующую информации:

- идентификатор текущего процесса;

- идентификатор текущего (main) потока;

- приоритет (nice) текущего потока;

- номера доступных процессоров. 

**Задание 05**

Разработайте консольное Linux-приложение OS05\_05 на языке С, выполняющее длинный цикл.

#include <stdio.h>

#include <unistd.h>

#include <stdlib.h>

#include <sys/resource.h>

#include <signal.h>

int main() {

// Получаем текущий приоритет (nice)

int current\_nice = getpriority(PRIO\_PROCESS, 0);

printf("Текущий nice: %d\n", current\_nice);

// Длинный цикл

printf("Запуск длительного цикла. Нажмите Ctrl+C для завершения.\n");

while (1) {

// Выполняем какую-то бесполезную работу

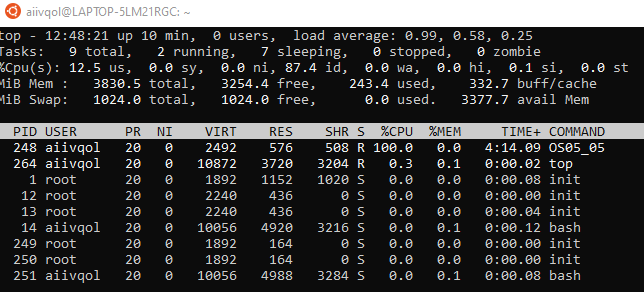
for (volatile int i = 0; i < 1000000; ++i);

}

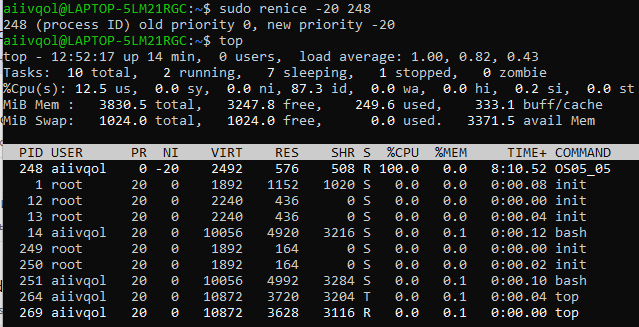
return 0;

}

Зафиксируйте текущее значение nicе, полученное с помощью команды top.



Увеличьте приоритет для OS05\_05 до максимального значения (самого привилегированного). Зафиксируйте текущее значение nicе, полученное с помощью команды top.



Уменьшите приоритет для OS05\_05 до минимального значения (самого ничтожного). Зафиксируйте текущее значение nicе, полученное с помощью команды top.

