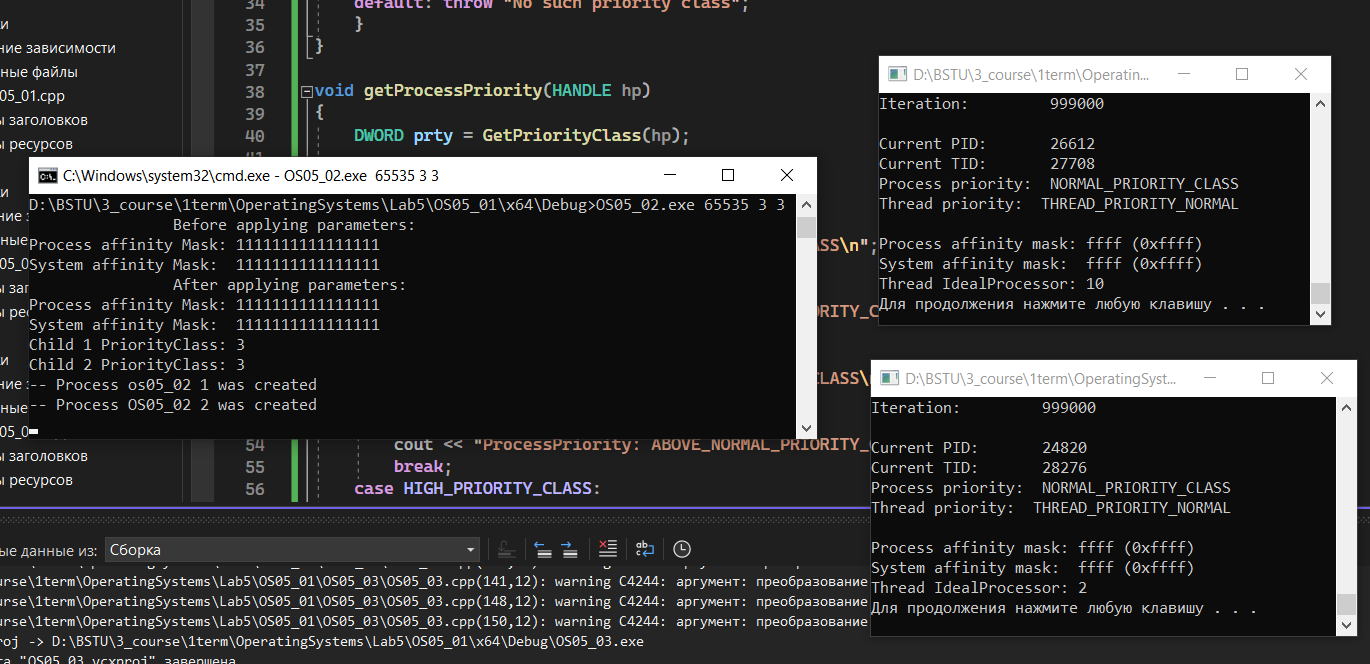
**Задание 2**

6. Запустите приложение **OS05\_02**,принимающее следующие значения параметров:

* P1: доступны все процессоры;
* P2: Normal;
* P3: Normal.



В видео нет смысла, они завершаются одинаково, т.к. везде normal

7. Запустите приложение **OS05\_02**,принимающее следующие значения параметров:

* P1: доступны все процессоры;
* P2: Below Normal;
* P3: High.

Видео HIGH-BELOW\_NORMAL

8. Запустите приложение **OS05\_02**,принимающее следующие значения параметров:

* P1: доступен один процессор;
* P2: Below Normal;
* P3: High.

Видео ONE-PROC\_HIGH-BELOW\_NORMAL

**Задание 3**

1. Запустите приложение **OS05\_03**,принимающее следующие значения параметров:

* P1: доступны все процессоры;
* P2: Normal;
* P3: Normal;
* P4: Normal;

OS05\_03.exe 65535 3 3 3

Видео 3-ALL\_PROC\_NORMAL

2. Запустите приложение **OS05\_03**,принимающее следующие значения параметров:

* P1: доступны все процессоры;
* P2: Normal;
* P3: Lowest;
* P4: Highest.

Видео 3-ALL\_PROC-HIGHEST-LOWEST

Запустите приложение **OS05\_03**,принимающее следующие значения параметров:

* P1: доступен один процессор;
* P2: Normal;
* P3: Lowest;
* P4: Highest.

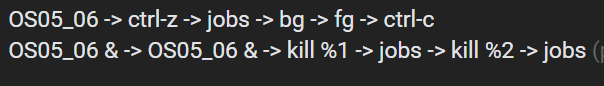
Видео 3-ONE\_PROC-HIGHEST-LOWEST

**Задание 5**

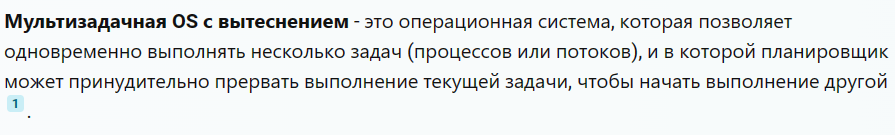
sudo renice –n -20 –p $(pgrep OS05\_05)

sudo renice –n 19 –p $(pgrep OS05\_05)

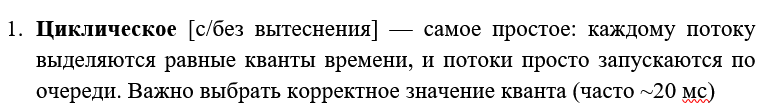
**Задание 6**



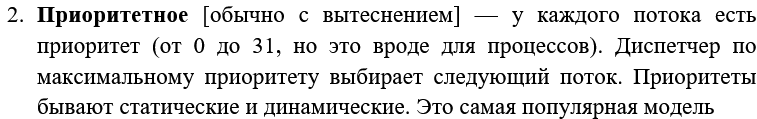
1. **Поясните понятие «мультизадачная OS с вытеснением».**

****

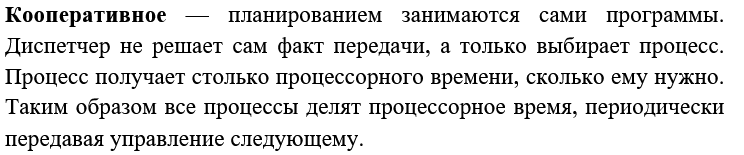
1. **Поясните понятие «циклическое планирование».**



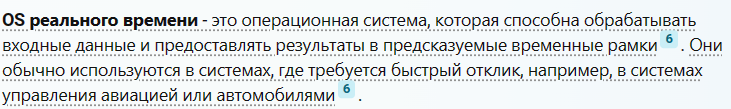
1. **Поясните понятие «приоритетное планирование».**



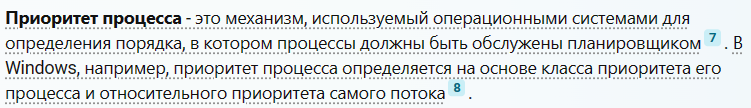
1. **Поясните понятие «кооперативное планирование».**



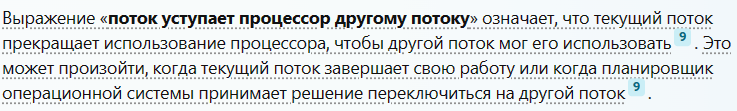
1. **Поясните понятие «OS реального времени».**

****

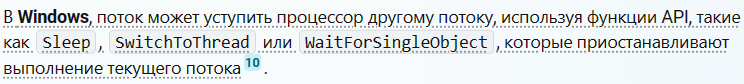
1. **Поясните понятие «приоритет процесса».**

****

1. **Поясните выражение «поток уступает процессор другому потоку».**

****

1. **Windows: как поток может уступить процессор?**

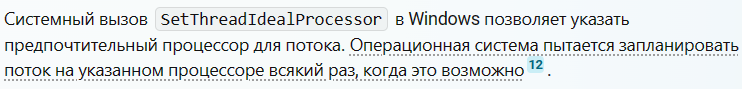
****

1. **Windows: что такое базовый приоритет потока, как он вычисляется и диапазон его изменения?**

Базовый приоритет — это то **значение**, чем будет являться приоритет потока с типом приоритета **Normal.**

****

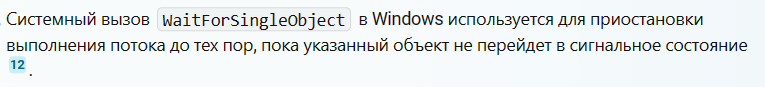
1. **Windows: поясните назначение и принцип применения системного вызова SetThreadIdealProcessor.**

****

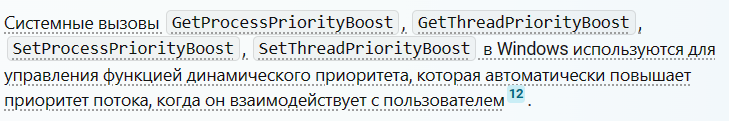
1. **Windows: поясните назначение и принцип применения системного вызова ResumeThread.**

****

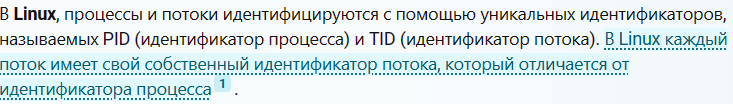
1. **Windows: поясните назначение и принцип применения системного вызова WaitForSingleObject.**

****

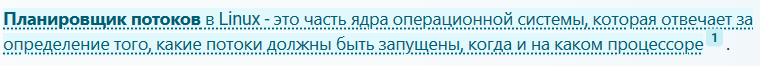
1. **Windows: поясните назначение и принцип применения системных вызовов GetProcessPriorityBoost, GetThreadPriorityBoost, SetProcessPriorityBoost, SetThreadPriorityBoost.**

****

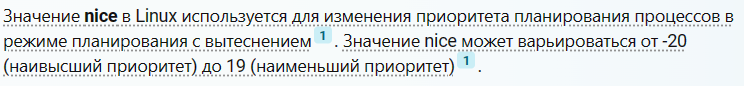
1. **Linux: поясните принцип идентификации процессов и потоков и поясните почему он такой.**

****

1. **Linux: Поясните понятие «планировщик потоков».**

****

1. **Linux: поясните принцип использования значения nice –процесса, диапазон его изменения, для какого режима работы планировщика это значение применяется?**

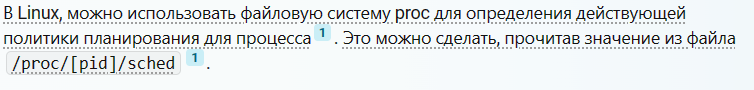
****

1. **Linux: перечислите политики планирования, какая действует по умолчанию?**

****

SCHED\_OTHER. SCHED\_RR, SCHED\_FIFO, SCHED\_BATCH

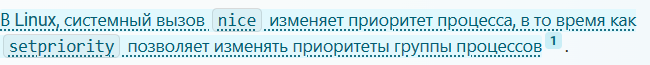
1. **Linux: как выяснить действующую политику планирования для процесса с помощью файловой системы proc?**

****

1. **Linux: с помощью какого системного вызова поток может уступить процессор.**

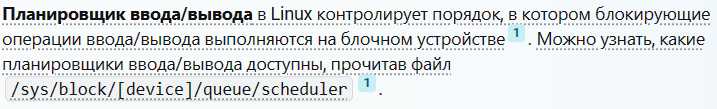


1. **Linux: чем отличается системный вызов nice от вызова setpriority.**

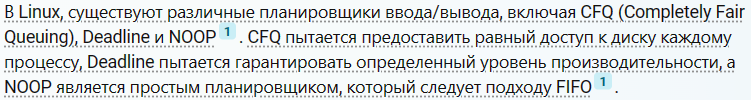
****

Процесс может поменять приоритет себе, используя системный вызов **nice**. Системные вызовы **getpriority, setpriority** определяют или меняют приоритеты любых процессов или групп процессов. Смотреть приоритеты чужих процессов может любой. Не суперпользователь может менять приоритеты только своих процессов, и не выше константы, которая называется NZERO. (Строго говоря, каждый пользователь имеет максимальный приоритет, выше которого ему нельзя устанавливать приоритеты своих процессов. Изменить максимальный приоритет любого пользователя может суперпользователь с помощью всё того же системного вызова **setpriority**).

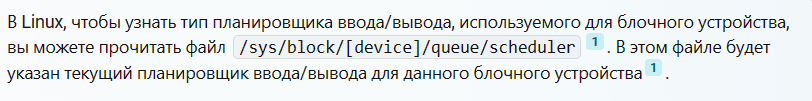
1. **Linux: поясните понятие «планировщик ввода вывода», каким образом можно выяснить какие планировщики ввода/ вывода доступны?**

****

1. **Linux: перечислите известные вам планировщики ввода/ вывода, кратко охарактеризуйте их.**

****

1. **Linux: каким образом можно выяснить тип планировщика действующего для блокового устройства?**

****