Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**Операционные системы**

Студент: Козека Е. М.

ФИТ 3 курс 4 группа

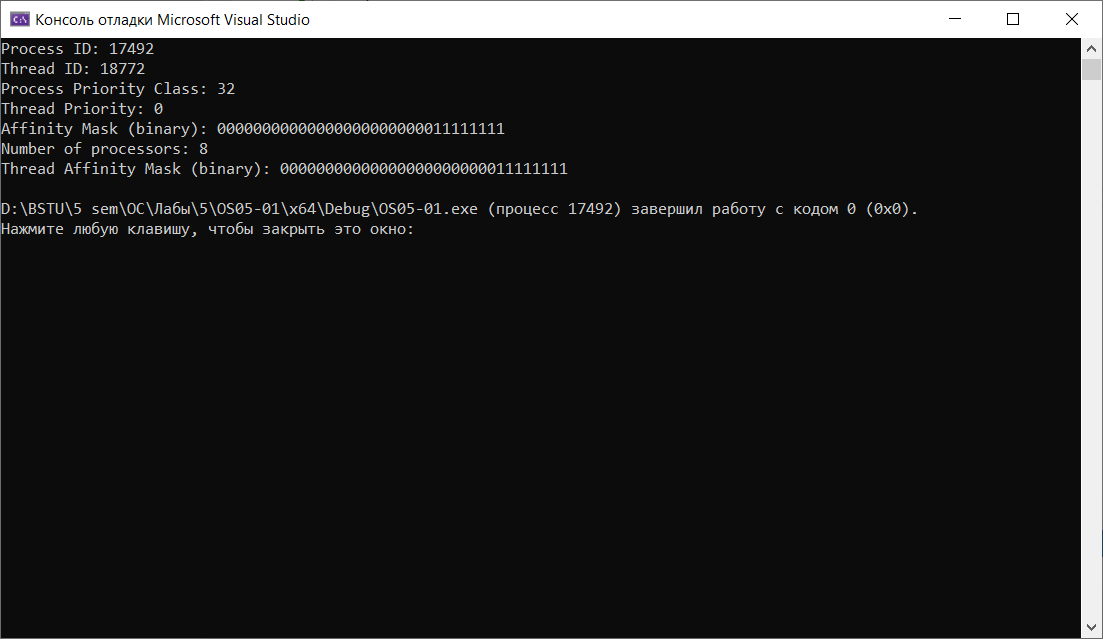
Преподаватель: Комкова А. В.

Минск 2024

**Лабораторная работа №5. Диспетчеризация**

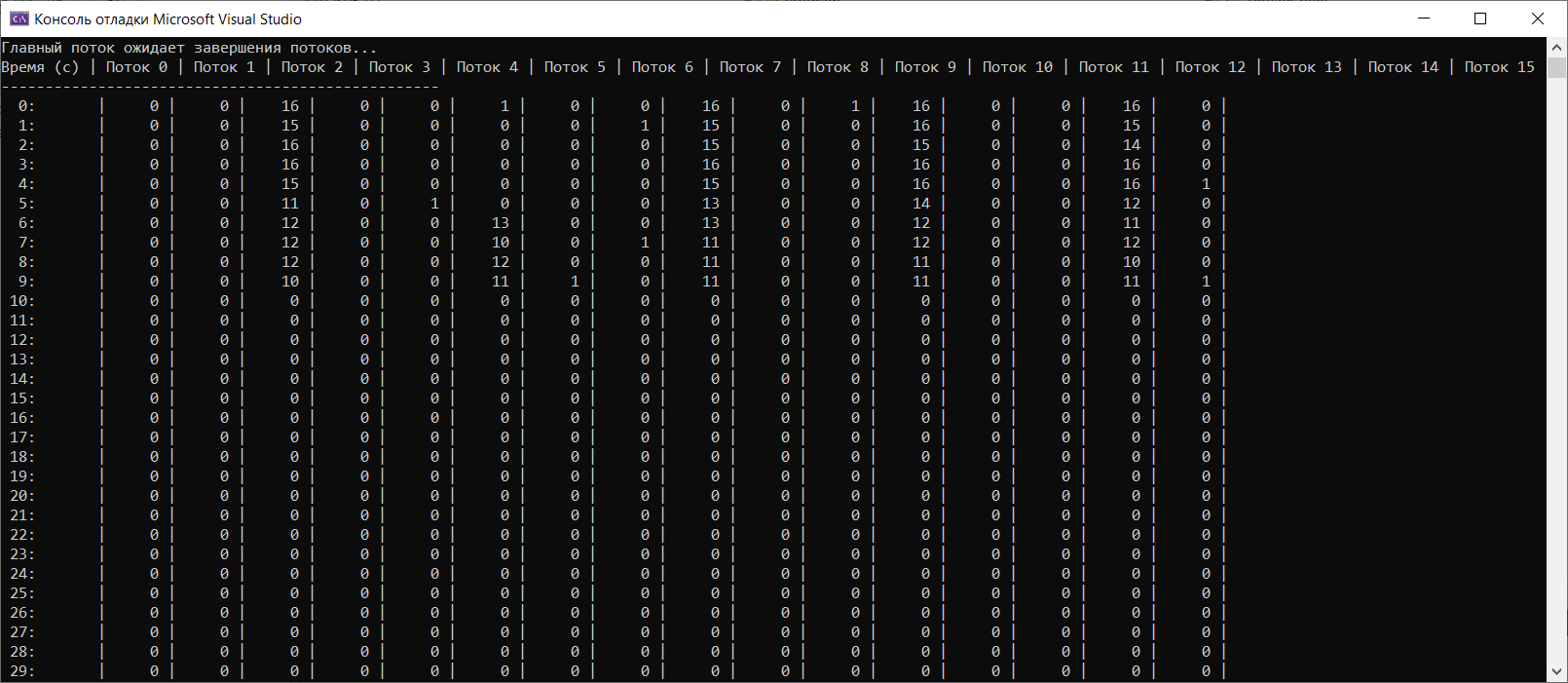
**Задание 1.** Разработайте консольное Windows-приложение OS05\_01 на языке С++, выводящее на консоль следующую информации: идентификатор текущего процесса; идентификатор текущего (main) потока; приоритет (приоритетный класс) текущего процесса; приоритет текущего потока; маску (affinity mask) доступных процессу процессоров в двоичном виде; количество процессоров, доступных процессу; процессор, назначенный текущему потоку.

Результат выполнения в консоли:



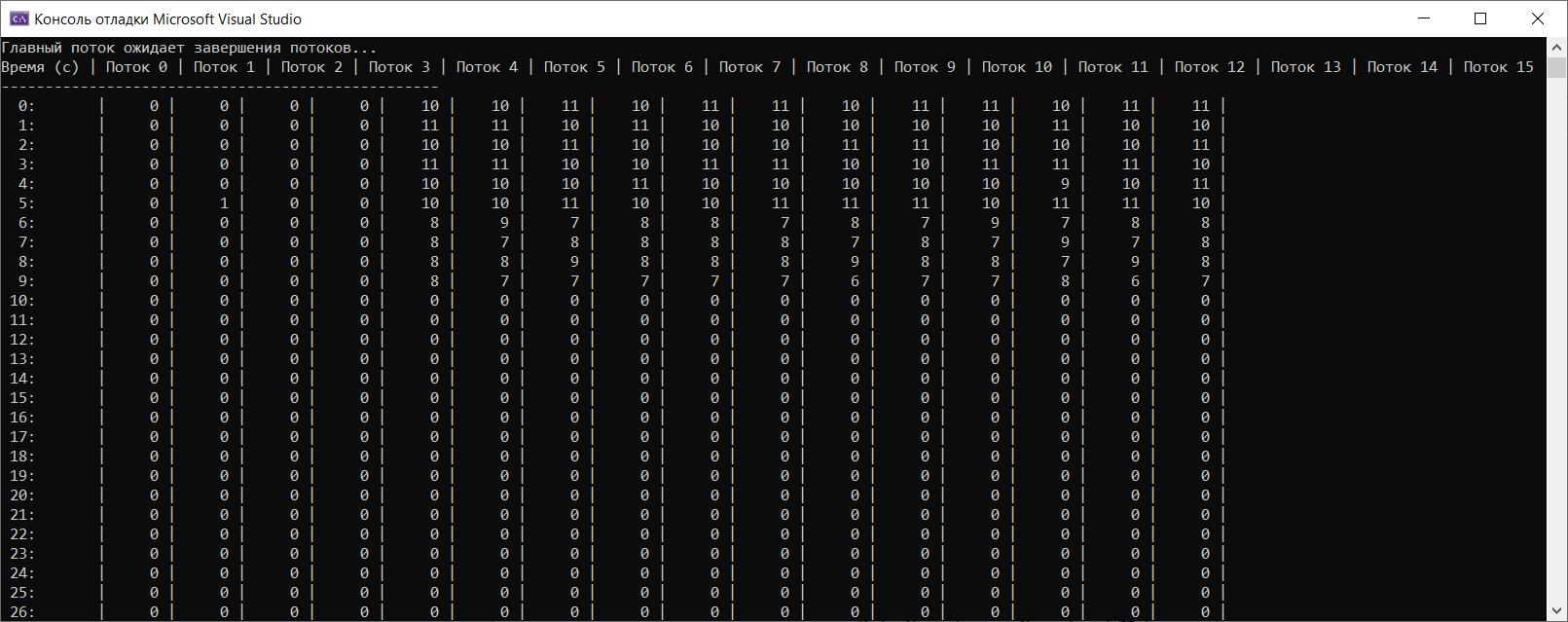
**Задание 2.** Создайте консольное Windows OS05\_02 на языке С#, взяв за основу приложение OS04\_07 из Лабораторной работы №4. Измените метод Main таким образом, чтобы потоки 0, 3, 6 и т.д. запускались с минимальным приоритетом потока, а потоки 2, 5, 8... – с максимальным. Класс приоритета процесса оставьте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

Результат выполнения в консоли:



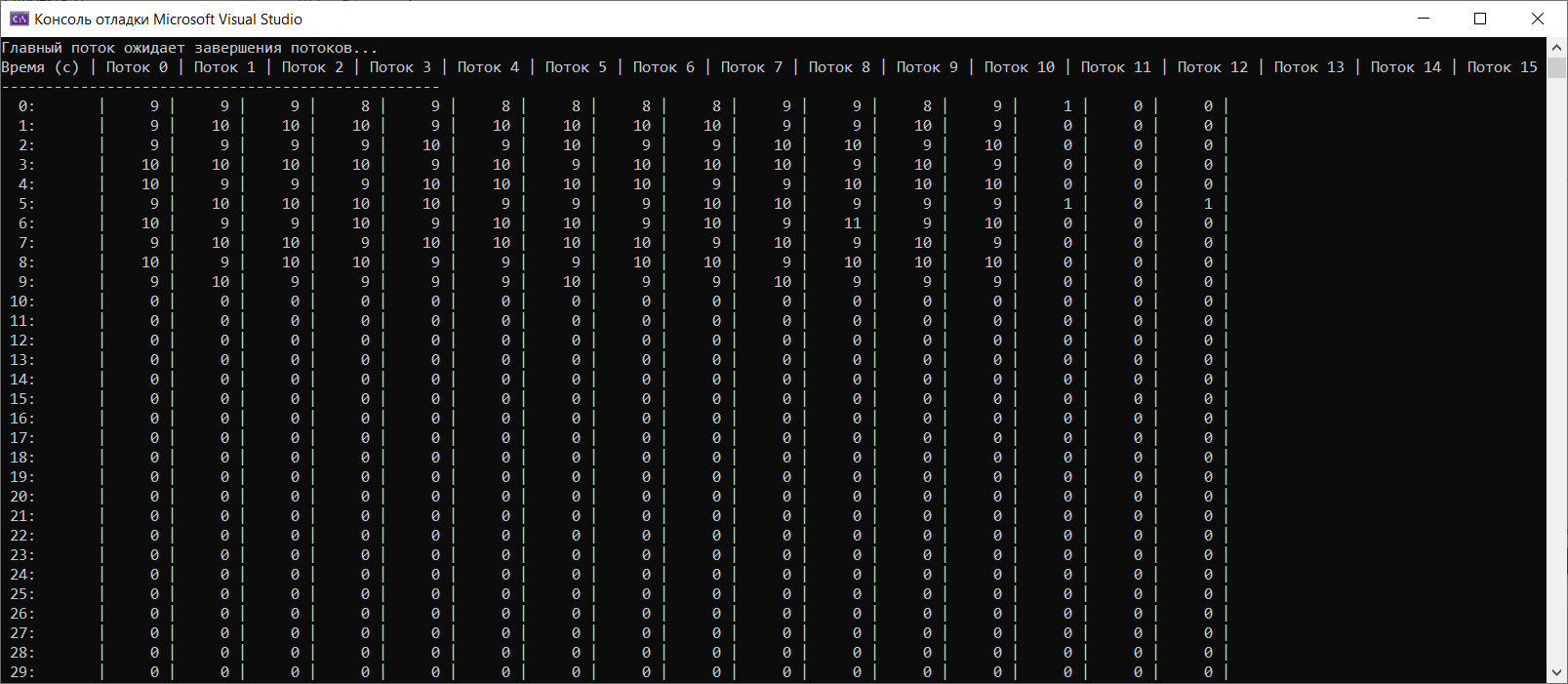
**Задание 3.** Создайте консольное Windows OS05\_03 на языке С#, взяв за основу приложение OS05\_02 из настоящей работы. На этот раз только несколько потоков запустите на наименьшем приоритете потока, а остальные – на наибольшем.

Результат выполнения в консоли:



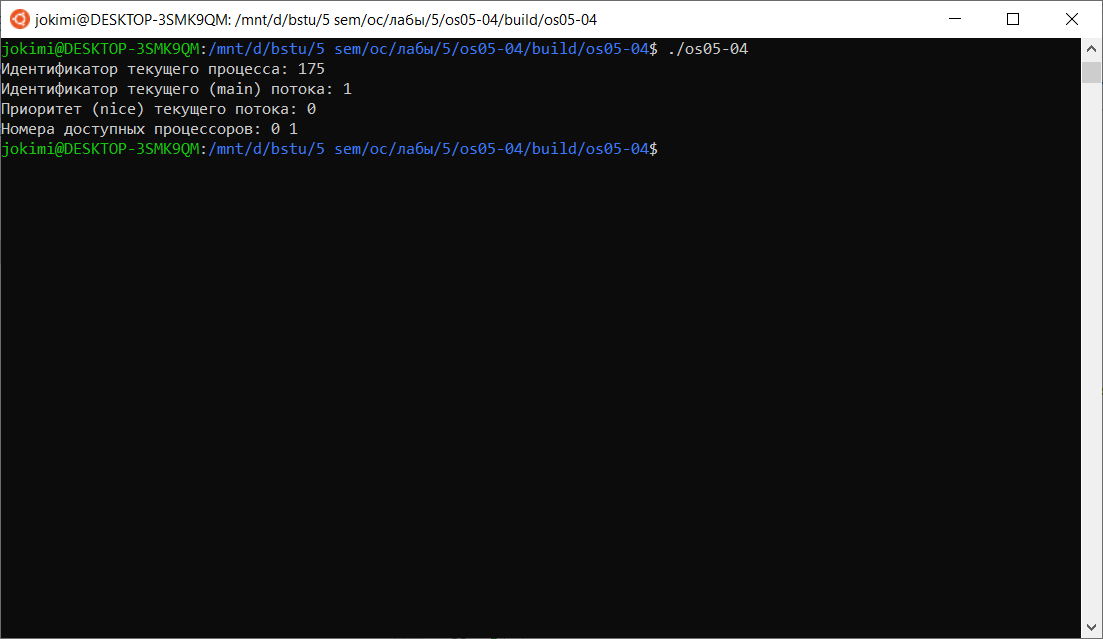
Значения для потоков с 0 по 3 значительно ниже, чем у потоков с высоким приоритетом, что подтверждает, что они получили меньше времени процессора.

Выполните приложение OS05\_03 с другими парами приоритетов, например, BelowNormal и Normal. Изменился ли характер работы потоков?



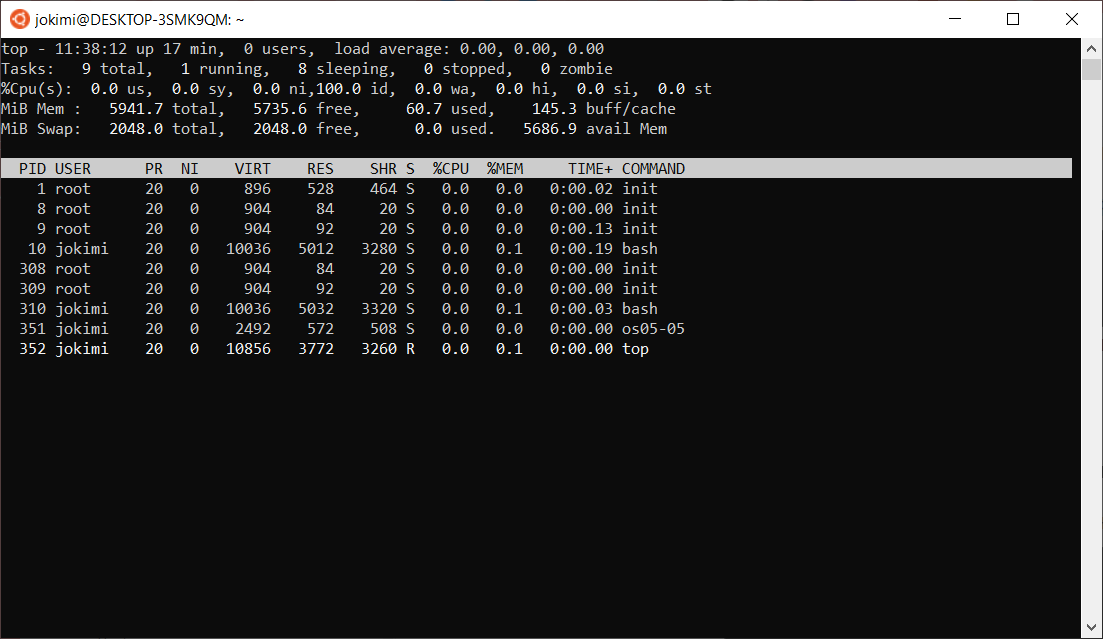
Потоки с приоритетом Normal (потоки 0-12) в большинстве случаев показывают стабильные значения в пределах 9-10, что говорит о том, что они активно выполняются. Значения для потоков 13-15 в основном колеблются в пределах 0-10, и с течением времени их активность снижается до 0, что означает, что эти потоки практически не выполняются.

**Задание 4.** 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.

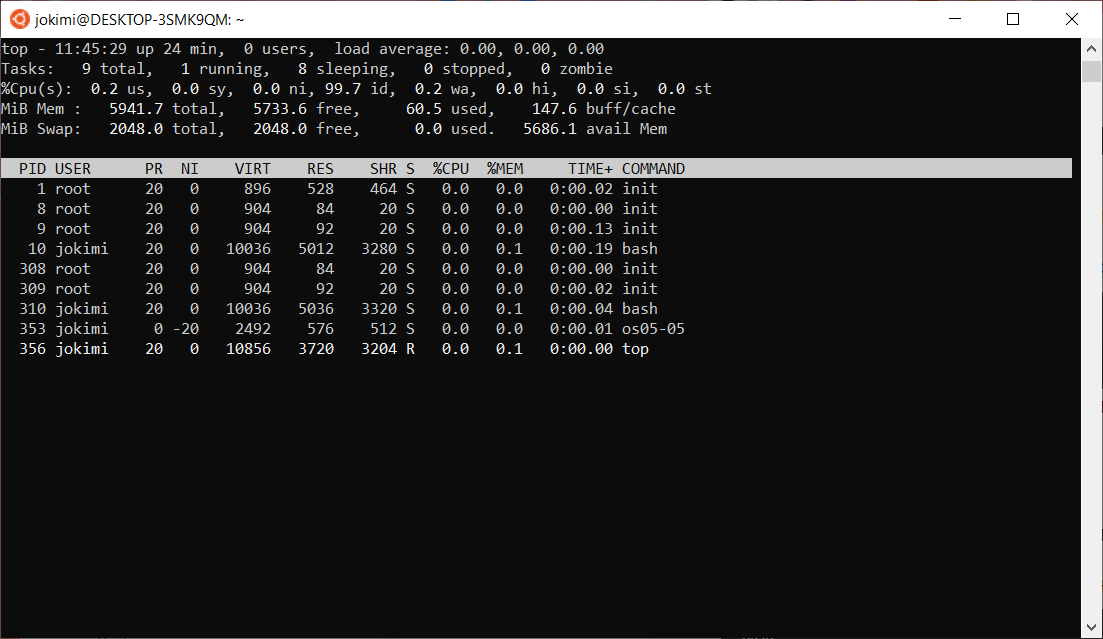


**Задание 5.** Разработайте консольное Linux-приложение OS05\_05 на языке С, выполняющее длинный цикл.

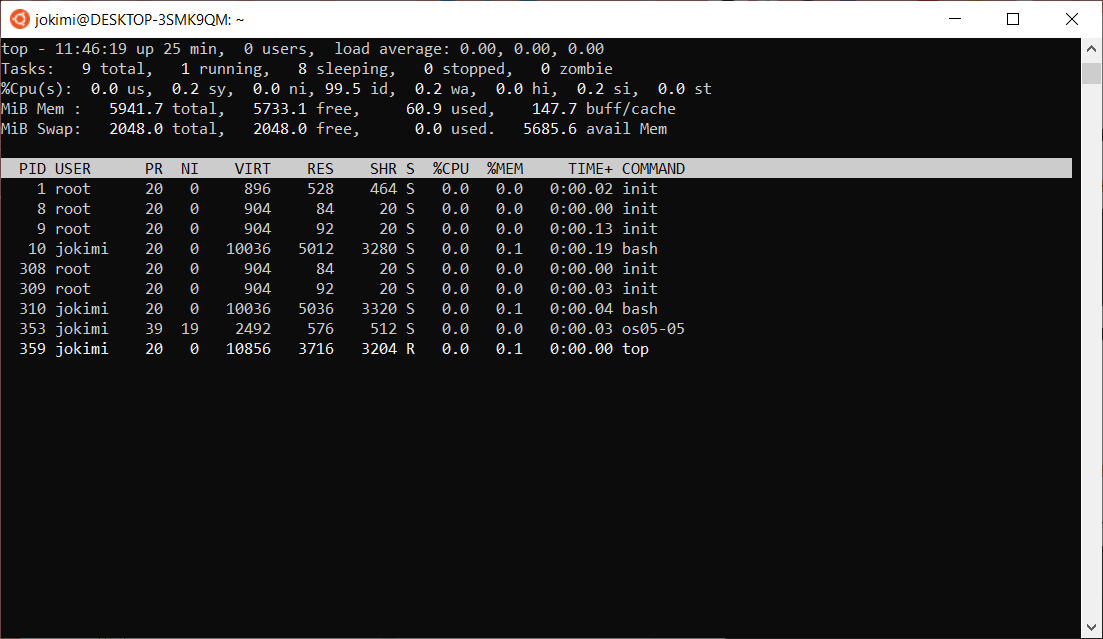
Зафиксируйте текущее значение nicе, полученное с помощью команды top.



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.



Ответы на вопросы:

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

ОС, которая выполняет несколько задач одновременно, выделяет квант времени, определяет, когда это время истекает и переключает задачу.

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

Каждому процессу назначается определенный интервал времени, называемый его квантом, в течение которого ему предоставляется возможность выполнения. Если процесс к завершению кванта времени все еще выполняется, то ресурс центрального процессора у него отбирается и передается другому процессу. Если процесс переходит в заблокированное состояние или завершает свою работу до истечения кванта времени, то переключение центрального процессора на другой процесс происходит именно в этот момент.

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

Каждому процессу присваивается значение приоритетности и запускается тот процесс, который находится в состоянии готовности и имеет наивысший приоритет.

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

Это такой алгоритм планирования, при котором процесс получает столько процессорного времени, сколько он считает нужным. Таким образом, все процессы делят процессорное время, периодически передавая управление следующей задаче.

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

Это система, которая гарантирует фиксированное время для выполнения задачи. Если операция должна быть проведена точно в срок (или в определенный период времени), то мы имеем дело с системой жесткого реального времени. Другой разновидностью подобных систем является система мягкого реального времени, в которой хотя и нежелательно, но вполне допустимо несоблюдение срока какого-нибудь действия, что не наносит непоправимого вреда.

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

Число, ориентируясь на значение которого планировщик процессов может выдавать процессу больше или меньше процессорного времени. Это значение, которое влияет на выбор процесса, который будет выполняться следующим.

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

Это значит, что текущий поток прерывает свое выполнение. При этом управление процессором передается следующему из очереди потоку.

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

Sleep(0) – процесс становится в конец очереди.

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

Базовый приоритет потока — сочетание класса приоритета процесса и приоритета потока, изменяется в пределах [1; 31], по умолчанию – 8, приоритеты возрастающие.

Вычисляется так:



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

SetThreadIdealProcessor устанавливает предпочтительный процессор для потока, т.е. процессор, на котором он будет работать.

DWORD SetThreadIdealProcessor(

HANDLE hThread, // дескриптор потока

DWORD dwIdealProcessor // номер идеального процессора

);

dwIdealProcessor — номер привилегированного процессора для потока. Значение MAXIMUM\_PROCESSORS сообщает системе, что поток не имеет привилегированного процессора.

Если функция завершается успешно, величина возвращаемого значения - предшествующий привилегированный процессор или MAXIMUM\_PROCESSORS, если поток не имеет такового. Если функция завершается с ошибкой, величина возвращаемого значения равна -1.

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

Запускает поток, после его системного вызова SuspendThread.

DWORD ResumeThread (

HANDLE hThread // Дескриптор потока

);

Уменьшает счетчик приостановленных потоков. Когда счетчик приостановки уменьшается до нуля, выполнение потока возобновляется.

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

DWORD WaitForSingleObject(

HANDLE hHandle, // Дескриптор потока

DWORD dwMilliseconds // Интервал ожидания в миллисекундах

);

Ожидает, пока указанный объект не перейдет в сигнальное состояние или пока не истечет интервал времени ожидания.

**13. Windows: поясните назначение и принцип применения системного вызова: GetProcessPriorityBoost, GetThreadPriorityBoost, SetProcessPriorityBoost, SetThreadPriorityBoost.**

GetProcessPriorityBoost, GetThreadPriorityBoost возвращает true/false в зависимости от того, разрешено ли динамическое изменение приоритета процесса/потока. SetProcessPriorityBoost, SetThreadPriorityBoost задают это разрешение.

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

Номер процесса всегда равен номеру главного потока, потому что ядро Linux не различает потоки и процессы.

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

Это алгоритм, который определяет порядок выполнения потоков.

Планировщик потоков ядра отвечает за то, какие потоки выполняются на процессорах системы. Каждый поток выполняется до тех пор, пока прерывание таймера не сигнализирует о том, что пора переключаться на другой поток (квант закончился), или до тех пор, когда потоку нужно ждать какого-то события (завершения ввода-вывода или снятия блокировки), либо до тех пор, пока работоспособным не станет поток с более высоким приоритетом (которому требуется процессор). При переключении с одного потока на другой планировщик обеспечивает сохранение регистров и прочего состояния оборудования. Затем планировщик выбирает для выполнения на процессоре другой поток и восстанавливает ранее сохраненное состояние (для выбранного потока).

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

nice — значение любезности, чем меньше значение, тем выше приоритет. Диапазон приоритетов — [-20, 19], default = 0. Чтобы установить значение nice ниже нуля, требуются права суперпользователя.

РЕЖИМ РАБОТЫ: вытесняющий(other). Планировщик может принудительно забирать управление у потока (например, по таймеру или при появлении потока с большим приоритетом).

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

Linux, политики планирования процесса: 1) стандартная (OTHER, разделения времени); — по умолчанию; 2) FIFO-политика (FIFO, реального времени); 3) карусельная (round-robin) политика (RR); 4) пакетная политика(BATCH).

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

shed\_yield().