Лабораторная работа 04

OC

**Задание 01**

1. Разработайте консольное Windows-приложение **OS04\_01** на языке С++, выполняющее длинный цикл с временной задержкой и с выводом на консоль идентификаторов текущего процесса и текущего потока.
2. Продемонстрируйте информацию об потоках процесса **OS04\_01** с помощью утилит  **PowerShell ISE** и **Performance Monitor**.

**Задание 02**

1. Разработайте на языке консольное Windows-приложение **OS04\_02** на языке С++, выполняющее цикл 100 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса.
2. Процесс **OS04\_02** должен создать два потока: потоковые функции **OS04\_02\_T1, OS04\_02\_T2.**
3. Поток **OS04\_02\_T1** - выполняет цикл 50 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса и потока.
4. Поток **OS04\_02\_T2** - выполняет цикл 125 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса и потока.
5. Продемонстрируйте информацию об потоках процесса **OS04\_02** с помощью утилит  **PowerShell ISE** и **Performance Monitor**.

**Задание 03.**

1. Разработайте на языке консольное Windows-приложение **OS04\_03** на языке С++, выполняющее цикл 100 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса.
2. Процесс **OS04\_03** должен создать два потока: потоковые функции **OS04\_03\_T1, OS04\_03\_T2.**
3. Поток **OS04\_03\_T1** - выполняет цикл 50 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса и потока.
4. Поток **OS04\_03\_T2** - выполняет цикл 125 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса и потока.
5. Поток **main** приостанавливает работу потока **OS04\_03\_T1** на20й своей итерации и возобновляют на 60й своей итерации.
6. Поток **main** приостанавливает работу потока **OS04\_03\_T2** на40й своей итерации и возобновляют после окончания собственных итераций (по окончании собственного цикла).
7. Продемонстрируйте информацию об потоках процесса **OS04\_03** с помощью утилит  **PowerShell ISE** и **Performance Monitor**.

**Задание 04**

1. Разработайте на языке консольное Windows-приложение **OS04\_04** на языке С++, выполняющее цикл 100 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса.
2. Процесс **OS04\_04** должен создать два потока: потоковые функции **OS04\_04\_T1, OS04\_04\_T2.**
3. Поток **OS04\_04\_T1** - выполняет цикл 50 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса и потока. Поток засыпает на 25й собственный итерации на 10сек.
4. Поток **OS04\_04\_T2** - выполняет цикл 125 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса и потока. Поток засыпает на 80й собственной итерации на 15сек.
5. Поток **main** засыпает на 30й итерации на 10 сек.
6. Продемонстрируйте информацию об потоках процесса **OS04\_04** с помощью утилит  **PowerShell ISE** и **Performance Monitor**.

**Задание 05**

1. Разработайте на языке консольное Windows-приложение **OS04\_05** на языке С++, выполняющее цикл 100 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса.
2. Процесс **OS04\_05** должен создать два потока: потоковые функции **OS04\_05\_T1, OS04\_05\_T2.**
3. Поток **OS04\_05\_T1** - выполняет цикл 50 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса и потока.
4. Поток **OS04\_05\_T2** - выполняет цикл 125 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса и потока.
5. Поток **main** завершаетпоток **OS04\_05\_T2** на 40й собственной итерации.
6. Продемонстрируйте информацию об потоках процесса **OS04\_05** с помощью утилит  **PowerShell ISE** и **Performance Monitor**.

**Задание 06**

1. Разработайте на языке консольное Linux-приложение **OS04\_06** на языке С, выполняющее длинный цикл с временной задержкой и с выводом на консоль идентификатора процесса.
2. Продемонстрируйте информацию о потоках процесса **OS04\_06** с помощью утилиты **ps**.

**Задание 07**

1. Разработайте на языке консольное Linux-приложение **OS04\_07** на языке С, выполняющее цикл 100 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса.
2. Процесс **OS04\_07** должен создать поток: потоковая функция **OS04\_07\_T1.**
3. Поток **OS04\_07\_T1** - выполняет цикл 75 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса.
4. Продемонстрируйте информацию о потоках процесса **OS04\_07** с помощью утилиты **ps**.

**Задание 08**

1. Разработайте на языке консольное Linux-приложение **OS04\_08** на языке С, выполняющее цикл 100 итераций с временной задержкой в 1 сек. с выводом на консоль идентификатора процесса.
2. Процесс **OS04\_08** должен создать поток: потоковая функция **OS04\_08\_T1.**
3. Поток **OS04\_08\_T1** - выполняет цикл 75 итераций с временной задержкой в 1 сек. с выводом на консоль идентификаторов процесса. Поток должен засыпать на 10сек. на 50й собственной итерации.
4. Поток **main** должен засыпать на 15 сек. на 30й собственной итерации.
5. Продемонстрируйте информацию о потоках процесса **OS04\_08** с помощью утилиты **ps**.

**Задание 09.ответьте на следующие вопросы**

1. **Что такое поток управления OS?**

Объект ядра операционной системы, которому ОС выделяет процессорное время, наименьшая единица работы ядра ОС.

1. **С помощью каких системных вызовов создаются потоки в Windows и Linux?**

Windows: Потоки создаются с помощью функций WinAPI, таких как CreateThread.

Linux: В Linux потоки можно создавать с помощью системных вызовов, таких как pthread\_create.

1. **Что такое системные и пользовательские потоки?**

Системные потоки управляются операционной системой и создаются и завершаются с ее помощью. Пользовательские потоки управляются прикладной программой и не зависят от операционной системы.

1. **Что такое многопоточность?**

Это концепция, при которой программа делится на несколько потоков выполнения, каждый из которых может выполняться независимо и параллельно с другими потоками. Это позволяет улучшить эффективность и производительность программы, особенно на многоядерных процессорах.

1. **Чем отличаются приоритетная многопоточность от и кооперативной многопоточности?**

Приоритетная многопоточность позволяет операционной системе самостоятельно прерывать выполнение одного потока для переключения на другой с более высоким приоритетом.

Кооперативная многопоточность требует явного сотрудничества потоков, чтобы они передавали управление друг другу. Операционная система не вмешивается в планирование потоков в случае кооперативной многозадачности.

1. **Что такое диспетчеризация потоков управления OS?**

В общем, при работе нескольких потоков или процессов нужно определить, какой из них будет работать и какое количество времени. Именно этим и занимается диспетчеризация. Она позволяет эффективно загрузить процессор.

Диспетчеризация — это процесс переключения процессора с одного потока на другой соответственно плану.

1. **Что такое контекст потока и для чего он нужен?**

Это набор данных, необходимых для восстановления состояния потока при его переключении.

1. **Перечислите состояния в которых может быть поток и поясните их назначение.**

Running (Выполнение): Поток активен и выполняется на процессоре.

Ready (Готовность): Поток готов к выполнению, но ожидает своей очереди на процессор.

Blocked (Блокировка): Поток заблокирован и ожидает какого-то события или ресурса.

Terminated (Завершение): Поток завершил свою работу и освободил ресурсы.

1. **Что такое LWP?**

LWP - это легковесный процесс, который представляет собой поток выполнения внутри процесса.

1. **Что такое потокобезопасность программного кода?**

Потокобезопасность программного кода - это свойство кода быть безопасным при одновременном доступе из нескольких параллельных потоков выполнения, чтобы избежать конфликтов и ошибок при работе с общими данными.

1. **Что такое реентерабельность кода?**

Реентерабельность кода - это способность функции или участка кода быть безопасно вызванными несколько раз одновременно из разных потоков без возникновения конфликтов.

1. **Что такое Fiber?**

Fiber - это легковесный поток выполнения, управляемый самой программой, который обеспечивает более гибкий контроль над выполнением задач в сравнении с обычными потоками операционной системы.

1. **Дайте развернутое определение потока OS.**

Поток операционной системы (OS) - это независимая последовательность инструкций, которая может выполняться параллельно с другими потоками в операционной системе. Каждый поток имеет свой собственный стек вызовов и ресурсы, но может взаимодействовать с другими потоками в пределах общего процесса.