Министерство образования Республики Беларусь

Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Институт информационных технологий

Специальность ИПОИТ

**КОНТРОЛЬНАЯ РАБОТА**

По предмету «Операционные системы»

Студент-заочник 2 курса

Группы: №680971

ФИО: Пасюк Александр Игоревич

Тел.: +375(25)763-74-66

Проверил: Бакунов Александр Михайлович

Минск, 2017

**СОДЕРЖАНИЕ**

1. Процессы в операционных системах. Состояние. Операции над процессами.
2. Средствами ms-dos создать копию flash-диска. Сравнить оба диска. Отчет расшифровать.
3. **Процессы в операционных системах. Состояние. Операции над процессами.**

**Проце́сс** — программа, которая выполняется в текущий момент. Стандарт ISO 9000:2000 определяет процесс как совокупность взаимосвязанных и взаимодействующих действий, преобразующих входящие данные в исходящие.

В вычислительной технике, процесс является экземпляром компьютерной программы, которая выполняется. Он содержит программный код и его текущей деятельности. В зависимости от используемой операционной системы (ОС), процесс может быть составлен из нескольких потоков выполнения, которые выполняют инструкции одновременно.

Компьютерная программа сама по себе — это только пассивная совокупность инструкций, в то время как процесс — это непосредственное выполнение этих инструкций.

Также, процессом называют выполняющуюся программу и все её элементы: адресное пространство, глобальные переменные, регистры, стек, открытые файлы и т. д.

Компьютерная программа представляет собой пассивный сбор инструкций, в то время как процесс является фактическим выполнением этих инструкций. Несколько процессов могут быть связаны с той же самой программой; например, открывая несколько экземпляров одной и той же программы часто означает нечто большее, чем выполнение одного процесса .

Метод многозадачности позволяет нескольким процессам совместно использовать процессоры (CPU) и другие системные ресурсы. Каждый процессор выполняет одну задачу за один раз. Тем не менее, многозадачность позволяет каждому процессору переключаться между задачами, которые выполняются без необходимости ждать каждой задачи, чтобы закончить. В зависимости от реализации операционной системы, коммутаторы могут быть выполнены, когда задачи выполнения операций ввода / вывода, когда задача указывает на то, что он может быть включен, или на аппаратных прерываний. Распространенной формой многозадачности с разделением времени. Время обмена является способ для обеспечения быстрого отклика для интерактивных пользовательских приложений. В разделения времени систем, переключение контекста выполняются быстро, что создает впечатление, что несколько процессов выполняется одновременно на одном процессоре. Эта кажущаяся выполнение нескольких процессов одновременно называется параллелизмом.

В целях обеспечения безопасности и надежности, большинство современных операционных систем предотвращения прямой связи между независимыми процессами, обеспечивают строго опосредованные и контролируемые функциональные возможности межпроцессного взаимодействия.

В общем, компьютерная система процесса состоит из следующих ресурсов: Образ исполняемого машинного кода, связанного с программой.

Память вычислительной технике, процесс является экземпляром компьютерной программы, которая выполняется. Он содержит программный код и его текущей деятельности. В зависимости от используемой операционной системы (ОС), процесс может быть составлен из нескольких потоков выполнения, которые выполняют инструкции одновременно. В вычислительной технике, процесс является экземпляром компьютерной программы, которая выполняется. Он содержит программный код и его текущей деятельности. В зависимости от используемой операционной системы (ОС), процесс может быть составлен из нескольких потоков выполнения, которые выполняют инструкции одновременно. ( как правило, некоторая область виртуальной памяти ); который включает в себя исполняемый код, данные конкретного процесса (вход и выход), то стек вызовов (для отслеживания активных подпрограмм и / или других событий), и кучи для хранения данных промежуточных вычислений, генерируемые во время выполнения.

Операционная система дескрипторы ресурсов, которые выделяются для процесса, таких как дескрипторов файлов ( Unix терминологии) , или ручками ( для Windows ), а также источников данных и стоков. Безопасность атрибуты, такие как владелец процесса и процесса "набор разрешений (допустимые операции).

Процессор состояние ( контекст ), такие, как содержание регистров и физической адресации памяти. Состояние, как правило, хранится в компьютерных регистров, когда процесс выполняется, и в памяти иным образом.

Операционная система удерживает большую часть этой информации о активных процессов в структурах данных, называемых управления технологическими процессами блоков. Любое подмножество ресурсов, как правило, по меньшей мере, состояние процессора, может быть связан с каждым из процесса " нитей в операционных системах, которые поддерживают потоки или ребенка (дочь) процессов.

Операционная система сохраняет свои процессы разделения и выделяет ресурсы, необходимые им, чтобы они были менее склонны мешать друг другу и вызывать системные сбои (например, тупиковых или обмолота ). Операционная система может также обеспечивать механизмы межпроцессного взаимодействия для того, чтобы процессы взаимодействия в безопасных и предсказуемым образом.

Многозадачной операционной системы может просто переключаться между процессами , чтобы дать появление многих процессов исполняющий одновременно (то есть, параллельно ), хотя на самом деле только один процесс не может быть выполнение в любой момент времени на одноядерном процессоре (если не используется многопоточность или другие аналогичные технологии).

Это обычно связать один процесс с основной программой, а также дочерние процессы с любым спин-офф, параллельные процессы, которые ведут себя подобно асинхронными подпрограммами. Процесс называется собственных ресурсов, из которых образ своей программы (в памяти) является одним из таких ресурсов. Однако в многопроцессорных системах многие процессы могут работать от, или совместно, один и тот же возвратный программу в том же месте в памяти, но каждый процесс называется собственный свой собственный образ программы.

Процессы часто называют "задачи" в встраиваемых операционных систем. Смысл "процесса" (или задачи) "что - то , что занимает время", в отличие от "памяти", которая является "то , что занимает место". Приведенное выше описание относится к обоим процессам под управлением операционной системы, а также процессы , как это определено исчислений процессов .

Если процесс запрашивает что-то, для которого он должен ждать, он будет заблокирован. Когда процесс находится в заблокированном состоянии , он имеет право на обмен на диск, но это прозрачно в виртуальной памяти системы, где области памяти процесса может быть на самом деле на диске , а не в основной памяти в любой момент времени. Обратите внимание , что даже неиспользованные части активных процессов / задач (выполнения программ) , имеют право на обмен на диск. Все части выполняемой программы и ее данные не должны быть в физической памяти для связанного процесса быть активными.

**1.2 Состояние**

В многозадачной (многопроцессной) системе процесс может находиться в одном из трех основных состояний:

ВЫПОЛНЕНИЕ - активное состояние процесса, во время которого процесс обладает всеми необходимыми ресурсами и непосредственно выполняется процессором;

ОЖИДАНИЕ - пассивное состояние процесса, процесс заблокирован, он не может выполняться по своим внутренним причинам, он ждет осуществления некоторого события, например, завершения операции ввода-вывода, получения сообщения от другого процесса, освобождения какого-либо необходимого ему ресурса;

ГОТОВНОСТЬ - также пассивное состояние процесса, но в этом случае процесс заблокирован в связи с внешними по отношению к нему обстоятельствами: процесс имеет все требуемые для него ресурсы, он готов выполняться, однако процессор занят выполнением другого процесса.

В ходе жизненного цикла каждый процесс переходит из одного состояния в другое в соответствии с алгоритмом планирования процессов, реализуемым в данной операционной системе.

В состоянии ВЫПОЛНЕНИЕ в однопроцессорной системе может находиться только один процесс, а в каждом из состояний ОЖИДАНИЕ и ГОТОВНОСТЬ - несколько процессов, эти процессы образуют очереди соответственно ожидающих и готовых процессов. Жизненный цикл процесса начинается с состояния ГОТОВНОСТЬ, когда процесс готов к выполнению и ждет своей очереди. При активизации процесс переходит в состояние ВЫПОЛНЕНИЕ и находится в нем до тех пор, пока либо он сам освободит процессор, перейдя в состояние ОЖИДАНИЯ какого-нибудь события, либо будет насильно "вытеснен" из процессора, например, вследствие исчерпания отведенного данному процессу кванта процессорного времени. В последнем случае процесс возвращается в состояние ГОТОВНОСТЬ. В это же состояние процесс переходит из состояния ОЖИДАНИЕ, после того, как ожидаемое событие произойдет.

Работы системы ядра , что позволяет многозадачности нуждается в процессы , чтобы иметь определенные состояния . Названия этих состояний не стандартизированы, но они имеют аналогичные функциональные возможности .

Во-первых, процесс "создан" загружается из вторичного хранения устройства ( жесткого диска с , CD-ROM и т.д.) в основную память . После этого процесс планировщика присваивает его "ожидания" состояние. В то время как процесс «ждет», он ждет планировщик , чтобы сделать так называемый переключение контекста и загрузить процесс в процессор. Затем состояние процесса становится "работает", а процессор выполняет инструкции процесса.

Если процесс должен ждать ресурса (ждать ввода пользователя или файл, чтобы открыть, например), ему присваивается "заблокированный" состояние. Состояние процесса изменяется обратно на "не дожидаясь", когда процесс больше не нужно ждать.

После того как процесс завершает выполнение, или завершается операционной системой, он больше не нужен. Процесс удаляется немедленно или перемещается в "прекращена" состояние. Когда удаляется, он просто ждет , чтобы быть удалены из основной памяти.

Когда процессы общаются друг с другом, что называется "межпроцессного взаимодействия" (IPC). Процессы часто должны взаимодействовать, например, в трубопроводе оболочки, выход первого процесса необходимо перейти ко второму, и так далее к другому процессу. Предпочтительно, в хорошо структурированном виде не использования прерываний.

Это возможно даже для двух процессов, которые будут работать на разных машинах. Операционная система (ОС) , может отличаться от одного процесса к другому, поэтому некоторые медиатор (ы) ( так называемые протоколы ) необходимы.

К началу 1960 - х годов, программное обеспечение компьютерного управления эволюционировала от программного обеспечения для управления монитором , например IBSYS , к программному обеспечению исполнительного управления . Со временем компьютеры стали быстрее в то время как компьютер время был еще ни дешевым , ни в полной мере; такая среда сделала многозадачности возможным и необходимым. Мультипрограммирование означает , что несколько программ работать одновременно . Во - первых, более чем одна программа работала на одном процессоре, в результате основной однопроцессорной компьютерной архитектуры, и они разделили скудные и ограниченные аппаратные ресурсы; следовательно, параллелизм был последовательного характера. На более поздних системах с несколькими процессорами , несколько программ могут работать одновременно в параллель .

Программы состоят из последовательностей инструкций для процессоров. Один процессор может выполнять только одну инструкцию за один раз: нельзя запускать несколько программ одновременно. Программа может понадобиться ресурс , например , как устройство ввода, которое имеет большую задержку, или программа может начать какие - то медленные операции, такие как отправка выход на принтер. Это привело бы к процессору быть "простаивает" (не используется). Для того, чтобы держать процессор занят все время, выполнение такой программы прекращается и операционная система переключает процессор для запуска другой программы. Для пользователя, будет казаться, что эти программы работать одновременно (отсюда термин «параллельный»).

Вскоре после этого понятие "программа" была расширена понятием "выполняемой программы и ее контекст». Понятие процесса рождения, который также стал необходимым с изобретением Реентрантная кода. Темы пришли несколько позже. Тем не менее, с появлением таких понятий, как в режиме разделения времени, компьютерных сетей и многопроцессорных разделяемой памяти компьютеров, старый "многозадачности" уступили истинную многозадачность, многопроцессорность и, позже, многопоточность.

Некоторые современные процессоры сочетают два или более независимых процессоров в многоядерной конфигурации и может выполнять несколько процессов одновременно. Другой метод, называемый одновременной многопоточности (используется в Intel 's Hyper-Threading технология) может имитировать одновременное выполнение нескольких процессов или потоков. Задачи и процессы относятся в основном к тому же объекту. И, хотя они имеют несколько иные терминологические истории, они стали использоваться в качестве синонимов. На сегодняшний день термин процесс , как правило , предпочтительнее задачу, за исключением того, когда речь идет о "многозадачности", так как альтернативный термин "многопроцессорная", слишком легко спутать с многопроцессорной (который представляет собой компьютер с двумя или более процессорами).

**1.3. Операции над процессами**

Процесс самостоятельно не переходит из одного состояния в другое. Изменением состояния процессов занимается операционная система. Операции над процессами производимые операционной системой удобно объединить в пары:

- создание процесса – завершение процесса;

- приостановка процесса – запуск процесса;

- блокирование процесса – разблокирование процесса;

- изменение приоритета процесса.

Операции создание и завершение процесса являются одноразовыми, так как применяются к процессу не более одного раза (некоторые системные процессы при работе вычислительной системы не завершаются никогда). Остальные операции, связанные с изменением состояния процессов, являются многоразовыми. Одноразовые операции приводят к изменению количества процессов, находящихся под управлением операционной системы и всегда связаны с выделением или освобождением определенных ресурсов. Многоразовые операции не приводят к изменению количества процессов и не обязаны быть связанными с выделением или освобождением ресурсов. Одноразовая операция создание процесса. Сложный жизненный путь процесса начинается с его рождения. В простых системах все процессы могут порождаться на этапе старта системы, в более сложных системах – создаваться динамически по мере необходимости. Инициатором рождения процесса после старта операционной системы может быть процесс пользователя, совершивший специальный системный вызов, либо процесс созданный самой операционной системой. Процесс, создавший новый процесс, называют процессом родителем, а созданный процесс – потомком (дочерним процессом или ребенком). Выделяют четыре основных события, которые приводят к созданию новых процессов:

- загрузка (инициализация) системы;

- выполнение изданного работающим процессом системного запроса на создание процесса;

- запрос пользователя на создание процесса;

- инициирование пакетного задания.

При загрузке операционной системы создаются несколько процессов, одни – высокоприоритетные, обеспечивающие взаимодействие с пользователем и выполняющие заданную работу, другие – фоновые, выполняющие особые функции. Например, один фоновый процесс обрабатывает приходящую электронную почту, активизируясь только при появлении писем, другой – обрабатывает запросы к web-страницам, активизируясь для обслуживания полученного запроса. Фоновые процессы, связанные с электронной почтой, web-страницами, новостями, выводом на печать и т.п. называются демонами.

2 Средствами ms-dos создать копию flash-диска. Сравнить оба диска. Отчет расшифровать.

Запустить MS-DOS.

1. d:\ms-dos>cmd
2. d:\ms-dos>xcopy d:\ms-dos\ms2



1. d:\ms-dos>fc /b readme.txt readme1.txt



