УВАЖАЕМЫЕ СТУДЕНТЫ! Изучите приведенную лекцию, законспектируйте основные понятия, дайте ответы на контрольные вопросы.

Ответы на вопросы, фотоотчет, предоставить преподавателю на e-mail: r.bigangel@gmail.com до 27.03.2023.

При возникновении вопросов по приведенному материалу обращаться по следующему номеру телефона: (072)111-37-59, (Viber, WhatsApp), vk.com: https://vk.com/daykini

ВНИМАНИЕ!!! При отправке работы, не забывайте указывать ФИО студента, наименование дисциплины, дата проведения занятия (по расписанию).

Лекция

Процессы. Создание, завершение процесса. Иерархии, состояния процессов. Контекст и дескриптор процесса.

План:

1. Понятия: задание, процесс, планирование процесса.
2. Состояния существования процесса.
3. Диспетчеризация процесса.
4. Блок состояния процесса.
5. Алгоритм диспетчеризации.
6. Понятие события.

1. Понятия: задание, процесс, планирование процесса.

Процесс - это программный модуль, выполняемый в CPU. Операционная система контролирует следующую деятельность, связанную с процессами:

1. создание и удаление процессов
2. планирование процессов
3. синхронизация процессов
4. коммуникация процессов
5. разрешение тупиковых ситуаций

Планирование - обеспечение поочередного доступа процессов к одному процессору.

Планировщик - отвечающая за это часть операционной системы. Алгоритм планирования - используемый алгоритм для планирования. Ситуации, когда необходимо планирование:

1. Когда создается процесс
2. Когда процесс завершает работу
3. Когда процесс блокируется на операции ввода / вывода, семафоре, и т.д.
4. При прерывании ввода / вывода.

Понятие «Процесс».

Не следует смешивать понятия процесс и программа. Программа - это план действий, а процесс - это само действие.

Понятие процесс, включает:

1. программный код
2. данные
3. содержимое стека
4. содержимое адрессного и других регистров CPU.

Таким образом, для одной программы могут быть созданы несколько процессов, в том случае, если с помощью одной программы в компьютере выполняется несколько несовпадающих последовательностей команд. За время существования процесс многократно изменяет свое состояние.

Различают следующие состояния процесса:

1. новый (new, процесс только что создан)
2. выполняемый (running, команды программы выполняются в CPU)
3. ожидающий (waiting, процесс ожидает завершения некоторого события, чаще всего операции ввода - вывода)
4. готовый (ready, процесс ожидает освобождения CPU)
5. завершенный (terminated, процесс завершил свою работу)

Каждый процесс представлен в операционной системе набором данных, называемых process control block . В process control block процесс описывается набором значений, параметров, характеризующих его текущее состояние и используемых операционной системой для управления прохождением процесса через компьютер.

Планирование процессов.

Система управления процессами обеспечивает прохождение процесса через компьютер. В зависимости от состояния процесса ему должен быть предоставить тот или иной ресурс. Например, новый процесс необходимо разместить в основной памяти, следовательно, ему необходимо выделить часть адресного пространства. Процессу в состоянии готовый должно быть предоставлено процессорное время. Выполняемый процесс может потребовать оборудование ввода - вывода и доступ к файлу.

Распределение процессов между имеющимися ресурсами носит название планирование процессов. Одним из методом планирования процессов, ориентированных на эффективную загрузку ресурсов, является метод очередей ресурсов. Новые процессы находятся во входной очереди, часто называемой очередью работ - заданий (job queue).

Входная очередь располагается во внешней памяти, во входной очереди процессы ожидают освобождения ресурса - адресного пространства основной памяти. Готовые к выполнению процессы располагаются в основной памяти и связаны очередью готовых процессов или ready queue. Процессы в этой очереди ожидают освобождения ресурса процессорное время.

Процесс в состоянии ожидания завершения операции ввода - вывода находится в одной из очередей к оборудованию ввода - вывода, которая носит название devices queue. При прохождении через компьютер процесс мигрирует между различными очередями под управлением программы, которая называется планировщик. (scheduler) Операционная система, обеспечивающая режим мультипрограммирования, обычно включает два планировщика - долгосрочный (long term scheduler) и краткосрочный (short term scheduler / CPU scheduler).

Основное отличие между долгосрочным и краткосрочным планировщиками заключается в частоте запуска, например: краткосрочный планировщик может запускаться каждые 100 мс, долгосрочный - один раз за несколько минут.

Долгосрочный планировщик решает, какой из процессов, находящихся во входной очереди, должен быть переведен в очередь готовых процессов в случае освобождения ресурсов памяти.

Долгосрочный планировщик выбирает процесс из входной очереди с целью создания неоднородной мультипрограммной смеси. Это означает, что в очереди готовых процессов должны находиться в разной пропорции как процессы, ориентированные на ввод - вывод, так и процессы, ориентированные на преимущественную работу с CPU.

Краткосрочный планировщик решает, какой из процессов, находящихся в очереди готовых процессов, должен быть передан на выполнение в CPU. В некоторых операционных системах долгосрочный планировщик может отсутствовать. Например, в системах разделения времени (time sharing system), каждый новый процесс сразу же помещается в основную память.

Планирование процессора.

Краткосрочный планировщик выбирает процессы из очереди готовых процессов и передает их на выполнение в CPU. Существуют различные

алгоритмы или стратегии решения этой задачи, отличающиеся отношением к критериям планирования.

2. Состояния существования процесса.

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. Типичный граф состояний процесса показан на рисунке 1.

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.

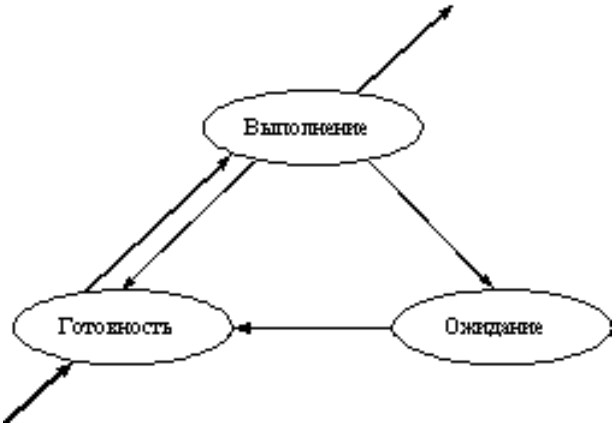

Рисунок 1 - Граф состояний процесса в многозадачной среде

3. Диспетчеризация процесса.

Диспетчеризация заключается в реализации найденного в результате планирования (динамического или статистического) решения, то есть в переключении процессора с одного потока на другой.

Диспетчеризация сводится к следующему:

- сохранение контекста текущего потока, который требуется сменить;
- загрузка контекста нового потока, выбранного в результате планирования;
- запуск нового потока на выполнение.

4. Блок состояния процесса.

Несмотря на то, что процесс является независимым объектом, со своим счетчиком команд и внутренним состоянием, существует необходимость взаимодействия с другими процессами. Например, выходные данные одного процесса могут служить входными данными для другого процесса.

В зависимости от относительных скоростей процессов (скорости зависят от относительной сложности программ и процессорного времени, предоставляемого каждому процессу), может получиться, что процесс уже готов к запуску, но входных данных для него еще нет. В этом случае процесс блокируется до поступления входных данных.

Процесс блокируется, поскольку с точки зрения логики он не может продолжать свою работу (обычно это связано с отсутствием входных данных, ожидаемых процессом). Также возможна ситуация, когда процесс, готовый и способный работать, останавливается, поскольку операционная система решила предоставить на время процессор другому процессу.

На рисунке представлена диаграмма состояний, показывающая три возможных состояния процесса:

1. **Работающий** (в этот конкретный момент использующий процессор).
2. **Готовый к работе** (процесс временно приостановлен, чтобы позволить выполняться другому процессу).

3. **Заблокированный** (процесс не может быть запущен прежде, чем произойдет некое внешнее событие).

С точки зрения логики первые два состояния одинаковы. В обоих случаях процесс может быть запущен, только во втором случае недоступен процессор. Третье состояние отличается тем, что запустить процесс невозможно, независимо от загруженности процессора.

Рисунок 2 -

1. Процесс блокируется, ожидая входных данных
2. Планировщик выбирает другой процесс
3. Планировщик выбирает этот процесс
4. Доступны входные данные

Процесс может находиться в рабочем, готовом и заблокированном состоянии. Стрелками показаны возможные переходы между состояниями

5. Алгоритм диспетчеризации.

Алгоритм диспетчеризации, или дисциплина обслуживания заявок, также оказывает большое влияние на длительность пребывания заявок в памяти ЦВМ. Ниже предполагается, что процессы поступления и обслуживания заявок являются независимыми, и поступившая заявка немедленно начинает обслуживаться, если в этот момент ЦВМ свободна от выполнения других подпрограмм, вызванных поступившими ранее заявками. Если же в момент поступления очередной заявки ЦВМ занята обслуживанием некоторой поступившей ранее заявки, то в зависимости от типа поступившей и обслуживаемой заявки, а также от вида используемого алгоритма диспетчеризации, поступившая заявка может прервать выполняемую подпрограмму или же ожидать начала своего обслуживания в памяти ЦВМ. Выбранная заявка покидает очередь, но все еще продолжает оставаться в памяти ЦВМ вплоть до момента завершения процесса своего обслуживания.

ЦВМ *смешанные алгоритмы диспетчеризации вычислений*, отличающиеся друг от друга количеством испльзуемых уровней прерывания и числом потоков заявок, обслуживаемых на каждом уровне.

При разработке *алгоритмов диспетчеризации* необходимо предусматривать оптимальную организацию системы очередей для наиболее удобной их обработки. Здесь существуют две крайности: если очередь

упорядочена, то запрос извлекать из нее легко, но для осуществления вставки нового запроса приходится перебрать в среднем половину очереди; если очередь не упорядочена, ее нужно просматривать каждый раз для обслуживания очередного запроса, но процесс вставки нового запроса облегчен.

При использовании *алгоритмов диспетчеризации вычислений* с относительными приоритетами обслуживание каждой вновь поступившей заявки даже в самом благоприятном случае может быть начато лишь после завершения уже выполняющегося обслуживания предыдущей заявки, а в режиме пакетной обработки информации - только после полной ликвидации очереди обслуживаемых заявок, даже если эти заявки имеют меньший приоритет. Вследствие этого длительность пребывания в памяти ЦВМ некоторых наиболее важных заявок может оказаться недопустимо большой.

Учитывая специфику *алгоритмов диспетчеризации участка* станков, для реализации задач диспетчеризации необходимо создание специализированной аппаратуры сбора и обработки первичной информации, так как существующая аппаратура предназначена для ввода и обработки технико-экономической информации с помощью носителей.

Последующий анализ *алгоритмов диспетчеризации вычислений* в мультипрограммных управляющих ЦВМ проводится, в основном, для пуассоновских входных потоков заявок, так как именно для потоков этого вида в большинстве случаев удается получить достаточно простые для расчетов аналитические результаты. Однако при этом следует иметь в виду, что такие потоки являются предельным случаем потоков Эрланга и создают в классе этих потоков наиболее тяжелые условия для работы ЦВМ. Поэтому получаемые при пуассоновских потоках характеристики алгоритмов диспетчеризации являются верхними оценками для тех же характеристик при эрланговских потоках более высоких порядков. Аналогичные результаты при регулярных входных потоках заявок являются нижними оценками соответствующих характеристик, однако получение этих результатов аналитическими методами, как правило, оказывается очень сложным. Поэтому для исследования алгоритмов диспетчеризации при регулярных потоках заявок обычно используется метод статистического моделирования работы этих алгоритмов на универсальных ЦВМ.

6. Понятие события.

В операционных системах семейства Windows существует несколько журналов, в которые записывается большинство событий, происходящих в системе. При этом некоторые события записываются автоматически, а регистрацию других (в основном, касающихся безопасности системы) необходимо активировать и настраивать. Событиями можно назвать любые изменения состояния операционной системы. К ним относятся запуск системы (компьютера), вход в систему любого пользователя, любые процессы,

запускаемые или отключаемые в операционных системах, попытка доступа к системным файлам и защищенным файлам реестра и многое другое. События подразделяются на типы, к которым относятся:

1) Информация;
2) Предупреждение
3) ошибка.

Кроме этого, существуют еще такие события, как «*Успешный аудит*» и «*Неуспешный аудит*». Последние два типа событий предназначены для специалистов, обеспечивающих контроль безопасности системы, и в данной статье будут рассмотрены только в части, касающейся активации параметров «Журнала безопасности Windows».

События типа «**Информация**» (еще этот тип событий называется «Сведения») отображают факт успешной операции. Этим событиям соответствует пиктограмма в журнале событий в виде кружочка с буквой «i» внутри кружочка. Справа от пиктограммы текстом отображается тип события «Сведения».

Событие типа «**Предупреждение**» отображает некоторые проблемы, имеющие место в работе операционной системы. Данное событие свидетельствует о незначительной проблеме в работе системы (приложения) и не требует немедленного вмешательства пользователя, но регулярное появление одного и того же события может со временем привести к ошибкам. Этим событиям соответствует пиктограмма в журнале событий в виде желтого треугольника с восклицательным знаком внутри (не показано). Справа от пиктограммы текстом отображается тип события «Предупреждение».

Событие «**Ошибка**» отображает проблемы, которые могут привести к потере работоспособности системы или потере информации. Этим событиям соответствует пиктограмма в журнале событий в виде красного кружочка с восклицательным знаком внутри кружочка. Справа от пиктограммы текстом отображается тип события «Ошибка».