

ОСНОВНЫЕ ПОНЯТИЯ: ОПЕРАЦИОННАЯ СИСТЕМА, ПРОЦЕСС, ПОТОК. СПОСОБЫ ИХ РЕАЛИЗАЦИИ

Операционная система – это организованная совокупность программ, решающая определённый круг задач, предоставляющая пользователю широкий набор средств, позволяющий упростить управление, программирование, отладку и сопровождение программ, представляет собой интерфейс между пользователем и аппаратной частью машины.

Процесс – совокупность последовательных действий, необходимых для достижения какого-либо результата в вычислительной системе.

В современных операционных системах процесс реализуется как динамический объект, который имеет большое количество полей и методов, то есть атрибутов и сервисов. Он может находиться в больших количествах состояний; в отношении с другими объектами, т.е. иметь родственные классы, занимать или освобождать необходимые ему ресурсы. Процессы могут быть последовательными относительно друг друга, параллельными и комбинированными.

Процесс – логическая единица работы операционной системы.

Поток (нить) – сущность внутри процесса, отображающая одну из возможных подзадач процесса.

ОСНОВНЫЕ ПОНЯТИЯ: МНОГОЗАДАЧНОСТЬ, КООПЕРАТИВНАЯ МНОГОЗАДАЧНОСТЬ, МНОГОПОТОЧНОСТЬ, МУЛЬТИПРОЦЕССОРНАЯ ОБРАБОТКА

Многозадачность (мультипрограммирование) – способ организации вычислительного процесса, при котором различные потоки совместно используют процессор или процессоры (если их несколько).

Кооперативная многозадачность – потоками совместно используется не только процессор, но и другие ресурсы, например, оперативная память ОП.

Многопоточность – поддержка нескольких потоков внутри одного процессора. Она достигается, если количество процессоров больше или равно числу потоков (не всегда) (это приближённо, в реальной жизни по-другому).

Мультипроцессорная обработка – исполнение одного и того же кода операционной системы различными процессами, как на однопроцессорных, так и на многопроцессорных машинах. При этом возможно и мультипрограммирование, а на каждом из процессоров может попеременно выполняться закреплённый набор потоков.

РЕСУРСЫ

Ресурсы – различные средства и возможности, необходимые для организации и поддержки процессов. Операционная система должна обеспечить эффективный и бесконфликтный способ распределения ресурсов.

КЛАССИФИКАЦИЯ РЕСУРСОВ

Физические (реально существующие) и виртуальные. Виртуальные ресурсы не существуют в том виде, в котором они показывают себя пользователю. По существу, это некоторая модель физического ресурса, но построение его происходит всегда на основе физического ресурса.

Пассивные (выделили память, попользовались и освободили) и активные. Активные ресурсы могут выполнить действия по отношению к другим ресурсам.

Временные и постоянные.

КАТЕГОРИИ РЕСУРСОВ И ИХ ОСНОВНЫЕ ХАРАКТЕРИСТИКИ

Процессорное время

Память (Оперативная, виртуальная)

Периферийные устройства;

Математическое обеспечение (функции для работы с файлами, данными и т.д.)

ОСНОВНЫЕ ФУНКЦИИ ОПЕРАЦИОННЫХ СИСТЕМ

Главная задача операционной системы – это упростить общение пользователя с ЭВМ.

1 Управление процессами: диспетчеризация, планирование, синхронизация и взаимодействие процессов

2 Управление ресурсами: организация доступа, управление распределением памяти

3 Организация доступа, защита математического обеспечения

4 Управление периферийными устройствами

Функции по отношению к пользователю:

1 Организация интерфейса с одним или несколькими пользователями

2 Поддержка операционного окружения пользовательских задач

3 Обеспечение совместимости с другими операционными системами

4 Защита и безопасность информации (с точки зрения пользователя то, что ему важно)

ТИПЫ И СВОЙСТВА ОПЕРАЦИОННЫХ СИСТЕМ

Типы:

1 Операционные системы машин ЭВМ общего назначения (для широкого круга пользователей, высокая универсальность)

2 Операционные системы реального времени (ОСРВ) RT-Linux

Управление датчиками, широкий спектр этих средств, используются упрощённые алгоритмы, высокая надёжность.

3 Операционные системы мобильных или портативных устройств (машин)

4 Операционные системы специального назначения (бортовые, например, для самолётов, танков ...)

Свойства:

1 Надёжность. Операционная система должна быть надёжной, как и аппаратура, на которой она работает. На операционной системе должны быть средства определения, диагностирования и восстановления.

2 Защита. Ресурсы должны быть защищены от взаимного влияния задач пользователя, минимизация порчи программных данных.

3 Предсказуемость. Реакция операционной системы на запросы пользователя должна быть предсказуемой и не варьироваться.

4 Удобство. Должна быть учтена специфика пользователя.

5 Эффективность: при распределении и использовании ресурсов (это важно в первую очередь в операционных системах)

6 Гибкость. Настройка параметров, установка операционной системы пользователем вручную по умолчанию

7 Расширяемость. Подключение новых модулей без переустановки системы

8 Прозрачность. Открытая или закрытая операционная система.

ДЕСКРИПТОР ПРОЦЕССА

Дескриптор представляет собой информационную структуру, в которой сосредоточена управляющая информация, необходимая для планирования (или для планировщика) и управления процессами.

ВИДЫ ГРУПП ИНФОРМАЦИИ ДЕСКРИПТОРА

1. Информация по идентификации (она содержит уникальное имя или номер, чтобы система смогла их отслеживать и выполнять операции как над поименованными объектами)

2 Информация о ресурсах, которые необходимы процессу для его работы (конструктор процесса инициализирует его поле)

3 Информация о состоянии процесса. Эта информация позволяет определить текущее состояние и возможность перехода в следующее.

4 Информация о родственных связях (Она нужна, чтобы при использовании ресурсов родственника не произошло конфликтов) Используется для корректного завершения процессов.

5 Информация, которая необходима для учёта и планирования процессов. Здесь указываются ссылки на средства синхронизации между процессами, приоритет и др.

ВЗАИМОБЛОКИРОВКА

Взаимоблокировка – состояние, при котором все процессы не могут продолжить свою работу в силу бесконечного ожидания ресурсов, занятых самими этими процессами.

СПОСОБЫ ВОССТАНОВЛЕНИЯ РАБОТОСПОСОБНОСТИ ОПЕРАЦИОННОЙ СИСТЕМЫ

1 При помощи принудительной выгрузки ресурса

2 Путём уничтожения процесса (виновника)

3 Через откат (составляется вектор состояния) в предыдущую точку.

КЛАССИЧЕСКИЕ ДИСЦИПЛИНЫ ОБСЛУЖИВАНИЕ ОЧЕРЕДИ НА ИСПОЛНЕНИЕ ПРОЦЕССА

1 FIFO (First Input First Output) – дисциплина обслуживания в порядке поступления.

Она обеспечивает минимизацию дисперсии времени ожидания.

2 LIFO (Last Input First Output) – дисциплина обслуживания, обратная порядку поступления, является основой для построения стековой памяти.

Общим для FIFO и LIFO является то, что среднее время ожидания в очереди одинаково независимо от характеристик процесса. Например, если некоторые процессы предполагают длительное исполнение ресурсов, а другой наоборот, то и те, и другие в очереди пробудут в среднем одно и то же время.

ПРОБЛЕМЫ, ВОЗНИКАЮЩИЕ ПРИ ВЗАИМОДЕЙСТВИИ ПРОЦЕССОВ

При организации многопроцессорного взаимодействия и использования разделяемых ресурсов, возникает целый ряд проблем. Главным образом, эти проблемы связаны с обеспечением корректного доступа к разделяемым ресурсам. Выделяют две важнейшие задачи операционной системы: распределение ресурсов между процессами и организация корректного доступа (т.е. организация защиты ресурсов, выделенных определённому процессу, от неконтролируемого доступа со стороны других процессов). В частности, под организацией корректного доступа может пониматься требование, декларирующее необходимость обеспечения независимости работы параллельных процессов от порядка и интенсивности доступа этих процессов разделяемым ресурсам.

Если требование независимости работы параллельных процессов нарушается, то могут возникнуть ситуации, когда процессы конкурируют за разделяемый ресурс. Такие ситуации называются гонкой процессов, которые также являются проблемой многопроцессорного взаимодействия.

Единственный способ избежать гонок при использовании разделяющих ресурсов – контролировать доступ к любым разделяемым ресурсам в системе. При этом необходимо организовать взаимное исключение – т.е. такой способ работы с разделяемым ресурсом, при котором постулируется, что в тот момент, когда один из процессов работает с разделяемым ресурсом, все остальные процессы не могут иметь к нему доступ. Проблему взаимного

исключения сводится к тому, чтобы не допускать ситуации, когда два процесса одновременно находятся в критических секциях, связанных с одним и тем же ресурсом.

В свою очередь при организации взаимного исключения могут возникать проблемы блокировки и тупики.

Блокировка – это ситуация, когда доступ к разделяемому ресурсу одного из взаимодействующих процессов не обеспечивается из-за активности других, более приоритетных процессов.

Тупик (deadlock) – это ситуация, когда (из-за некорректной организации доступа и разделения ресурсов) конкурирующие за критический ресурс процессы вступают в клинч – происходит взаимоблокировка.

МЕХАНИЗМЫ СИНХРОНИЗАЦИИ И ВЗАИМОДЕЙСТВИЯ ПРОЦЕССОВ

В настоящий момент известны такие средства организации взаимодействия процессов (взаимного исключения) как семафоры Дейкстры, мьютексы и аппарат передачи сообщений.

Обмен сообщениями имеет сложную структуру. Асинхронность взаимодействия процессов усложняет взаимодействие. Это происходит от того, что один процесс выполняется со скоростью, независимой от скорости других процессов. Например, принимающий процесс может быть не готов в данный момент к приёму, в итоге чтобы обеспечить взаимодействие, используют временной буфер.

В итоге обмен между процессами может быть разделён на 2 класса: Первый класс – это разделяемые (совместно используемые) переменные (например, семафоры и события)

Событие – объект синхронизации, который используется для информирования потоков о том, что произошло с событием.

Семафор – счётчик числа доступных ресурсов, которые представляют собой неотрицательные целые переменные, используемые для синхронизации процессов. Для семафоров определены 2 операции p и v . Пусть переменная S – семафор, тогда $p(S)$ уменьшает S на единицу, если это возможно, так как меньше нуля быть не может. При этом прерывания запрещены, $v(S)$ – увеличивает на единицу.

Если уменьшить на единицу невозможно, то поток, вызывающий операцию p , ждёт пока это станет возможно. При этом операции уменьшения и проверки неделимы. Когда S может принимать значение только 0 или 1, то семафор S становится блокирующей переменной (двоичным семафором).

Использование двоичной переменной позволяет организовать доступ к критическому ресурсу только одному потоку, а семафор синхронизирует работу нескольких потоков.

Мьютексы (похожие на семафоры) используют всего 2 операции: захват и освобождение мьютекса. Основные отличия мьютекса от семафора:

Мьютекс может быть захвачен не более чем одним потоком управления. Такой поток называют владельцем мьютекса. Освободить мьютекс может только его владелец. Поэтому мьютексы нельзя использовать в функциях обработки прерываний.

Если мьютекс захвачен, то при попытке захвата другим потоком, захватывающий поток будет приостановлен и будет находиться в очереди к мьютексу. В этой очереди может находиться несколько потоков, упорядоченных по приоритету. Если потоков с одним приоритетом несколько, то они упорядочиваются по времени (простоянки) нахождения в очереди (первый тот, кто раньше пришёл)

В основном мьютексы используются для управления доступом к ресурсам, которые совместно используются разными потоками управления.

Операция захвата мьютекса позволяет обеспечить монопольный доступ к ресурсу. При использовании средств синхронизации может случиться, что менее приоритетные потоки будут мешать выполнению более приоритетных потоков. Такой феномен получил название инверсии приоритетов.

Мьютексы в отличие от семафоров позволяют избежать инверсию приоритетов.

Второй класс – сообщения. Используются в распределённых системах, когда процессы имеют собственную ОП, а использовать разделяемую память невозможно. Здесь возможны 3 подхода:

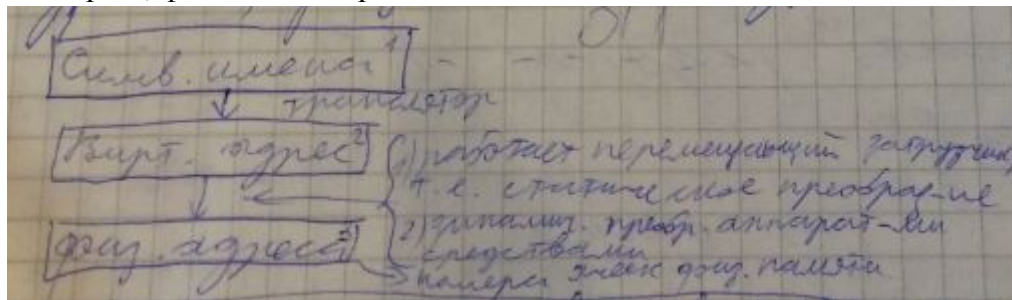
- 1) Процесс может обмениваться сообщением только с процессом, имеющим общего прародителя;
- 2) может обмениваться с любым процессом;
- 3) может обмениваться со взаимодействующими процессами.

ВИДЫ ПАМЯТИ

Физическая (память, способ адресации которой соответствует физическому расположению её данных)

Логическая (набор ссылок, которые может использовать программа)

Для идентификации переменных и команд используются символьные имена (метки), виртуальные адреса, физические адреса.



ОСНОВНЫЕ ФУНКЦИИ УПРАВЛЕНИЯ ОПЕРАТИВНОЙ ПАМЯТЮ

1) Отображение и перевод адресов логической памяти в физические адреса (т.е. распределение физических адресов от 0 до K среди m процессов). Выделяют следующие функции перевода:

А) абсолютная трансляция (выполняется компилятором или ассемблером при подготовке программы и генерирование ими абсолютных адресов)

Б) статическая трансляция – проводится, если программа составлена в форме выполняемого модуля: отдельные программы составляются (собираются) вместе, и им присваиваются адреса относительно некоторого фиксированного адреса памяти

В) Динамическая трансляция – когда реальные адреса расположения программы в памяти определяется операционной системой

Предполагается, что пространство памяти, отводимое процессу, может изменяться в ходе его работы.

2) расширение границы логического адресного пространства за границы физического адресного пространства.

Для этого используется оверлейное программирование (разные части программы используют одинаковый набор логических адресов). Существуют другие методы выхода за границы, такие как обмен (свопинг) и сцепление. При этих методах логическое адресное пространство любой программы соответствует физическому адресному пространству.

3) Разделение между программой пользователя и операционной системой и распределение (между ними).

4) Защита информации (пользователя от операционной системы и наоборот, друг от друга).

СВОПИНГ. ОСНОВНОЕ НАЗНАЧЕНИЕ И ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ

Эффективность использования оперативной памяти можно повысить с помощью внешней памяти, и наоборот. При нехватке оперативной памяти ненужные в данный момент

разделы могут копироваться на диск. Это называется откачкой. Обратный процесс – подкачка. А в совокупности эти процессы называются свопингом.

Для организации свопинга требуется планировщик памяти. Сложность свопинга в том, что один раздел может использоваться несколькими процессами (родственными, например).

- Свопинг позволяет заново распределить память для процесса, не запуская его с самого начала;
- Свопинг позволяет выполнить больше малоактивных процессов;
- Свопинг позволяет освободить память, занимаемую процессом, который требует вмешательство пользователя;
- Свопинг позволяет более эффективно использовать другие ресурсы, кроме оперативной памяти, например, при использовании приоритетов
- в многопользовательских системах, когда исполняется один и тот же код в свопинге могут участвовать только данные пользователя.

НАЗНАЧЕНИЕ И ЗАДАЧИ ВИРТУАЛЬНОЙ ОПЕРАТИВНОЙ ПАМЯТИ

Назначение: виртуальная память позволяет выполнять программы, требующие больше оперативной памяти, чем имеется в компьютере, путём автоматического перемещения частей программы между основной памятью и вторичным хранилищем (например, жёстким диском).

Задачи:

- 1) Размещение (нужно выбрать какие страницы выгрузить в ОП, а какие на диск). В адресном пространстве оперативной памяти выбираются страницы и/или сегменты, на которые будут отражаться некоторые страницы и/или сегменты виртуального пространства. Сложность: виртуальное пространство может быть намного больше физического. Цель – уменьшить фрагментацию.
- 2) Перемещение, например, из архивной среды хранения (из виртуальной среды) надо переместить информацию в оперативную память
- 3) Преобразование, то есть необходимо найти абсолютный адрес в рабочей среде хранения по его виртуальному адресу в соответствии с функцией преобразования (самый простой способ преобразования в фиксированные страницы)
- 4) Замещение, т.е. необходимо выбрать какую страницу или какой сегмент нужно выгружать, а на его место подставлять другой, т.е. выбрать кандидатов на перераспределение.

ФАЙЛОВАЯ СИСТЕМА

Файловая система – часть операционной системы, которая выполняет операции над файлами.

ЗАДАЧА И ФУНКЦИИ ФАЙЛОВОЙ СИСТЕМЫ

Задачи:

- 1) Обеспечение независимости программ от конкретной конфигурации машины;
- 2) Скрытие от пользователя реального расположения информации на физическом уровне
- 3) Осуществление доступа к данным

Функции:

- 1) Файловая система реализует метод организации данных;
- 2) Файловая система выполняет перевод логических характеристик в физические
- 3) Файловая система защищает пользователя от случайных сбоев оборудования
- 4) Файловая система позволяет многим пользователям разделять одно устройство.
- 5) Файловая система обеспечивает возможность восстанавливать данные
- 6) Файловая система должна защищать от несанкционированного доступа.

ХАРАКТЕРИСТИКИ ФАЙЛОВ

1 Тип файла (его структура):

1) Сегментированный – обеспечивает структуру файла с произвольным доступом. Может иметь неопределённый размер.

2) Непрерывный – обеспечивает отдельный непрерывный блок для быстрого и непосредственного доступа.

3) Последовательный – обеспечивает последовательную организацию и файл может расти неограниченно.

2 Имя файла:

В старых операционных системах (МС-ДОС 6.2 и ниже) использовали формат 8-3 и максимальная длина пути 80, а в современных ОС используются длинные имена до 255 символов и длина пути до 260 символов

3 Расширение файла:

Операционная система должна распознавать стандартный набор расширений.

4 Атрибуты файла:

Специфицируют тип файла, защиту и способ буферизации. Примеры:

- Пароль для доступа
- Владелец файла
- Скрытый
- Системный
- Архивный
- Только для чтения
- Длина
- Текущий размер