

Процесс - это: Набор команд, ресурс, момент исполнения.

Процесс и программа !=. Одно может состоять из нескольких частей другого.

Контекст- вся информация требуемая для процесса, сохраняется при приостановлении и восстанавливается при выделении ресурсов планировщиком

Части Контекста –

P C B Process Control Block – системный контекст, отвечает за планирование, управление памятью, средства ввода-вывода. Также включает в себя регистровый контекст – отвечает за регистры и счетчики.

Пользовательский контекст – отвечает за код и данные в адресном пространстве

Операции над процессами

Одноразовые – создание и завершение процесса

Многоразовые – запуск и приостановка, блокировка и разблокировка процессов. Изменение приоритета

Кооперация процесса

Кооперативные или взаимодействующие процессы – это процессы влияющие друг на друга, обмениваясь информацией

Причины для кооперации: повышение скорости, общий доступ данных, модульная конструкция, удобство.

Категории средств:

Сигнальные – минимум информации 1 бит «извещение»

Канальные – линии связи, ограничены пропускной способностью канала

Разделяемая память – область общего адресного пространства. Не требует дополнительных специальных вызовов. Самый быстрый и удобный способ в рамках одной машины.

Установка связи между процессами

Прямая – процессы общаются непосредственно между собой

При передаче данных явно указывается имя процесса источника и реципиента. Симметричная прямая адресация – если процессы указывают имена своих партнёров, в такую связь не может вмешаться ни один другой процесс. Если партнер не определен однозначно, то такая связь ассиметрично прямая. Можно использовать без дополнительных инициализаций, необходимо знать индикаторы процессов

Не прямая – данные передаются с помощью промежуточного объекта имеющего свой адрес, откуда процессы могут взять эти данные. Пример

доска объявлений, при этом процессы не знают об именах друг друга. Необходим индикатор промежуточного объекта.

Направленность связи

Однонаправленная связь (симплексная) – каждый процесс ассоциированный с ней может либо передавать либо принимать информацию.

Двунаправленная связь – процесс и принимает и передает. Полудуплексная – поочередная передача, Дуплексная – возможно одновременно передавать и получать.

Буферизация

Буфер нулевой емкости – никакая информация не сохраняется на линии связи, процесс донор должен ожидать пока инфу не примет реципиент (не применяется в жизни)

Буфер ограниченной емкости – буфер имеет размер n , если в буфере достаточно места, то процесс донор просто грузит инфу и не ждет. В ином случае донор останавливается пока реципиент не прочтет и не освободит буфер.

Буфер неограниченной емкости – теоретическое понятие, процесс донор никогда не ждет

Поток ввода-вывода и сообщения

Поток ввода-вывода – содержимое данных не важно. Данные простой поток байт Примеры

Pipe – трубка, может быть использована только в родственных процессах. Так как информацией о расположении pipe обладает только процесс создатель и потомки.

FIFO – именованный pipe, можно использовать для организации связи между любыми процессами

Сообщения – передаваемые данные в некоторой структуре. Поток разделяется на отдельные сообщения границами. Сообщения могут иметь фиксированный или произвольный размер..
Виды: очереди сообщений, socket.

Надежность средств связи – связь надежна если:

Нет потерь, Нет повреждений, Нет лишней инфы, Не нарушается порядок информации.

Нити исполнения (thread) – нити разделяют программный код, глобальные переменные и системные ресурсы. Но у каждой нити свой стек, содержимое регистров и программный счетчик
Традиционный процесс – процесс с одной нитью.
Нити могут создавать нити-потомки в рамках своего процесса. Состояния нити аналогичны состоянию процессов

Планирование процессов

Долгосрочное планирование – планирование заданий, отвечает за порождение новых процессов в системе и ее степень мультипрограммирования (кол-во процессов одновременно в системе)

Краткосрочное планирование – планирование использования процессора, обращение процесса к устройствам ввода вывода

Swapping – возвращение процесса на диск для выполнения в дальнейшем, для повышения эффективности, определяется промежуточным уровнем планирования – среднесрочным.

Цели планирования:

Занять 100% процессора, Сокращение (turnaround time) Общее время, (waiting time) время которое процессы в состоянии готовности, (response time) время отклика процесса.

Параметры планирования

Статические – предел ресурсов, Динамические – свободные ресурсы

CPU burst – время на процессоре. IO burst – время на ожидание

Невытесняющее планирование – процесс занимает сколько хочет, Возможен захват процессора 1 процессом

Вытесняющие планирование – процесс может быть прерван, а процессор передан другому

Алгоритмы планирования

F C F C – First Come First Served работают в том порядке в котором пришли

Robin Round RR – карусель процессов, закольцованный FCFC

Shortest job first S J F – вытесняющий и не вытесняющий приоритет у процесса с меньшим CPU burst

Приоритетное планирование – вытесняющее и не вытесняющий, по приоритету

Механизмы синхронизации

Семафоры – переменная принимающая не отрицательное значение, с только 2 атомарными операциями P проверка P(S): if s==0 wait else S-- и V увеличить. V(S): S++

Мониторы – подходят для ООП, имеет собственные переменные, может быть изменен методом монитора или совершить какое-то действие над состоянием. Важно, внутри монитора в любой момент времени может быть только 1 процесс

Сообщения – можно описать с помощью 2 примитивов send и receive

Прямая адресация – send(P, mess), receive(Q, mess)

Не прямая адресация - send(A mess), receive(A, mess)

A - адрес ящика

В пределах одного устройства медленнее чем семафор или монитор

Физическая организации памяти

Основная – оперативная, главная, массив однобайтовых ячеек с адресами

Вторичная память – одномерное линейное адресное пространство, энергонезависимая и имеет большую емкость

Иерархия памяти: Регистр – Кеш – Главная –

Электронные диски – Магнитные диски – Магнитная лента. Цена бита падает а время доступа и емкость увеличивается

Локальность – по мере снижения скорости доступа снижается частота доступа.

Принцип локальности – программы в течении какого-то промежутка работают с небольшим количеством адресов, выведен эмпирически

Свойство локальности - соседние в пространстве и времени объекты характеризуются похожими свойствами.

Физическое адресное пространство – адрес характеризующий реальное положение данных в физ памяти.

Логическая память – Адреса, к которым обращается процесс отличаются от адресов физических. Адреса могут быть представлены различными способами

Логическое адресное пространство – совокупность логических адресов

Связывание адресов – связывание логического и физического адреса. Связывание должно произойти до или в момент выполнения оператора. И может быть сделано по следующему:

Компиляция (Compile time) когда известно точное положение процесса, тогда генерируются физические адреса, но при изменении адреса старта нужно перекомпилировать

Загрузка (Load time) если нет инфы о расположении, компилятор генерирует перемещаемый код, связывание откладывается до загрузки программы. При перемещении страта надо только перезагрузить

Выполнение (Execution time) если процесс может быть перемещен во время выполнения то связывание откалывается до перемещения. Желательно наличие регистров перемещения

Функции системы управления памяти

Отображение адресного пространства на физическую память.

Распределение памяти между процессами

Контроль доступа к адресным пространствам процессов

Выгрузка процессов

Учет свободной и занятой памяти

Схемы управления памятью

Схема с фиксированными разделами – разделение на несколько разделов фиксированной величины.

Процессы распределяются по разделам. Подсистема оценивает размер процесса и выбирает подходящий раздел. Недостаток – число процессов ограничено числом разделов. Также минус то что модель страдает от внутренней фрагментации – потеря часть памяти которая была выделена но не используется процессом

Схема с переменными разделами – в данном случае раздел создаются только при поступлении новой задачи, для которой выделяется минимальное количество памяти. В конце память это куча разных разделов. Пустые смежные разделы могут быть объединены. Существуют следующие стратегии

First Fit – первый подходящий раздел

Best fit – в тот раздел который более как раз

Worst fit – в тот что менее всего как раз

Данный алгос страдает от внешней фрагментации – много не используемой и не выделенной памяти

Страничная память (paging) – память представляет собой страницы или блоки. При это образуются логические страницы которые соответствуют страничным кадрам (page frame), страницы имеют фиксированную длину и не перекрываются. Внешняя фрагментация отсутствует, а потери внутренней фрагментации ограничены только частью последней страницы

В данном случае логически адрес это $p+d$ где p – это номер страницы а d – смещение внутри страницы
Сегментная и сегментно страничная организация как страничная, но размер фрагментов переменный.
сегмент – линейная последовательность адресов.
Максимальный размер сегмента определяется разрядностью процессора. Размер сегмента может меняться динамически. Если размер смещения в виртуальном адресе выходит за пределы размера сегмента, возникает исключительная ситуация.

Логический адрес в данном случае номер сегмента и смещение внутри сегмента. Такой подход позволяет легко организовать совместное использование одних и тех же данных и программного кода разными задачами. логические блоки памяти разных процессов отображают в один и тот же блок физической памяти, где размещается разделяемый фрагмент кода или данных.

Виртуальная память – информация с которой работает активный процесс должны быть в оперативке. Память процесса разбивается на страницы, логический адрес динамически транслируется в физический. Если нужная страница не в физ памяти то ее нужно подкачать с диска, для этого в состав атрибутов страницы входит бит присутствия

Страничная виртуальная память как и обычная но подкачка целыми страницами только
Сегментно-страничная организации виртуальной памяти -вроде все тоже, но размер сегмента может быть больше оперативки

Структура таблицы страниц

Для страничной организации. Запись содержит номер кадра, атрибуты страницы: бит модификации, бит присутствия бит защиты

Ассоциативная память (translation lookaside buffer TLB) – быстрый кэш который обеспечивает хранение части таблицы. Это эффективно так как за частую 1 процесс использует небольшую часть таблицы

Аномалия belady – иногда увеличение кадров -> увеличение ошибок

Стратегии управления страничной памятью

Стратегия выборки (fetch policy) – (как переписать страницу в оперативку) выборка по запросу загрузка отсутствующей страницы в свободную физ память по запросу. Выборка с упреждением – опережающее чтение. Вместо 1 страницы подгружаются еще и соседние

Стратегия размещения (placement policy) – (какой участок физ памяти поместит в страницу) в любой свободный кадр.

Стратегия замещение (replacement policy) – (какую страницу вытолкнуть из оперативки) происходит с

учетом кол-ва кадров выделенного каждому процессу.

При замещении система должна: найти страницу на вытеснение, переместить ее на диск, записать на ее место страницу, модифицировать таблицу страниц

Алгоритмы замещения страниц

FIFO – для каждой страницы создается метка времени и в очередь

OPT – для каждой странице вычисляется количество команд до первого использования и наибольшая уходит. (Невозможный)

LRU - least recently used убирается тот который дольше всего не использовался. (Трудоемкий)

NFU(Not Frequently Used) – убирается самый редко использующийся. Требуем доп счетчики

Трешинг – высокая частота страничных нарушений

Процесс находится в состоянии трешинга, если при его работе больше времени уходит на подкачку страниц, нежели на выполнение команд

Модель рабочего множества – рабочее множество процесса $W(t, T)$ это набор страниц которые позволяет некоторое время работать с редкими page fault. Где t это момент времени а T параметр периода

Работа с файлами

Файловая система – часть ОС организующая эффективную работы с данными во внешней памяти, и интерфейс к ней

Индексация – прием разделения на блоки фиксированного размера, адреса блоков хранятся в отдельном блоке (индексе)

Основные функции файловой системы:

Идентификация файлов – связывания имени файла с пространством во внешней памяти

Распределение внешней памяти между файлами

Обеспечение надежности и отказоустойчивости
Обеспечение защиты от не санкционированного доступа

Обеспечение совместного доступа к файлам

Обеспечение высокой производительности

За частую имена файлов состоят из имени + расширения

Имена в разных каталогах могут совпадать. Для однозначного определения используют абсолютное имя (pathname)

Но это не всегда удобно, так что используют относительные пути

Типы файлов

Регулярные – пользовательская инфа

Директории – содержат структуру файловой системы

Атрибуты файлов – разные характеристики (дата, размер)

Структуризация файлов

(Простейший вариант) Последовательный файл – неструктурированная последовательность байтов

(На практике) Файлы прямого доступа – байты могут быть прочитаны в произвольном порядке

Устройства прямого (random) доступа – содержимое файла может быть разбросано по разным блокам диска

Представления файлов

Последовательности записей фиксированной длины – каждая запись имеет собственную структуру.

Операция чтения производится над записью, а операция записи переписывает или добавляет запись целиком

Последовательность записей переменной длины – каждая запись содержит ключевое поле. Базисная операция – считать запись по значению ключа, такой метод доступа индексно -последовательный

Операции над файлами

Создание – объявить файл и атрибуты,
выделить место

Удаление – освобождение памяти

Закрытие – освободить место внутренних
таблиц

Позиционирование –указать место чтение
внутри файла

Чтение – с указанием объёма или буфера

Запись – с конкретной позиции

Операции над директориями

Создание – с записями . и ..

Удаление – удалить можно только пустую дир.

Открытие – для чтения с получением всех имен
файлов

Закрытие – освобождение внутренних таблиц

Поиск – содержимое текущей записи в текущей
дир.

Получение списка файлов

Переименование

Создание файла – при создании нужно добавить
файл в каталог

Удаление файла – удаление из каталога

Алгоритмы обработки запросов к регистрам диска

FCFC (First come first serve) – в очереди FIFO

SSTF (Short Seek Time First) – обслуживание
ближайшего, при равенстве расстояний можно по
разному, например FCFC

SCAN – из стороны с сторону до обоих концов

LOOK – из стороны в сторону, но не доходя до конца
если не требуется

С SCAN – циклический скан при достижении конца
быстрое перемещение в другой конец

С Look – циклический look при достижении
последнего запроса быстрое перемещение в другой
конец ленты к первому запросу

Подсказки к задачам

Планирование процессора и памяти

Количество моментов времени — это сумма CPU burst строим таблицу размерами для каждого процесса моментов времени начиная с 1

Затем диаграмму памяти столбцами в количестве моментов времени.

Если нет места в памяти процесс ждет в очереди.

Процессы в очереди влияют на долгосрочное планирование, если пришел процесс нужного размера но с меньшим приоритетом он ждет

Turnaround time t_t – среднее время нахождения процесса в системе, время в очереди и время готовности тоже считаются

Waiting time среднее ожидание в очереди и готовности

Запросы к жесткому диску

Нарисовать штуку стрелочками

Считай очень аккуратно

Работа MMU

Вычисляем смещение внутри сегмента – степень сдвойки размера сегмента в байтах, а (адрес - количество битов смещения) = биты под номер страницы (они будут старшие)

Смещение внутри страницы – степень сдвойки размера страницы в байтах. (смещение сегмента - смещение страницы) = номер страницы (старшие)

Разбиваем логический вид на байты, допустим номер страницы это 9 бит а номер страницы 5 бит

Тогда 0x0010011111 0x28 0x10 первые 9 бит отсюда это номер и это 1 (если нет страницы такого номера то, это sef fault)

Смещения 0011111 0x28 0x10 = 0x1a2810

Проверить что смещение меньше чем длина сегмента, иначе (seg falt)

Далее 0011111 0x28 0x10 первые 5 бит это норме страницы и это 7, проверить что эта страница есть иначе page fault

Тогда смещение страницы 0x032810=11 0x28 0x10
Допустим что нашли кадр 22 (дес) = переводим в 0x16 22= 00010110

Физический адрес это 00010110+ 0x032810 = 0x005b2810

Это ответ

Ассоциативная память TLB

$t(\text{среднее}) = t_1 + h \cdot t_o + (1-h)(C+1)t_o$

где t_o – время доступа к оперативке

t_1 – время tlb

h – hit ration

C – количество уровней орагнизаци

Страничная память

Сначала последовательность сжимается

(повторяющиеся вызовы удаляются)

Затем рисуется таблица по горизонтали обращения к страницам а по вертикали номера свободных мест + строчка был ли page fault