

Лекция 3: Планирование процессов

Задача распределения ресурсов между потребителями -- планирование ресурсов: основываясь на поставленной цели, алгоритма и параметров.

Планирование заданий -- появляется, когда появилось возможность выбирать задание

Планирование использования процессора -- появляется в мультипрограммных системах.

Уровни планирования

То что раньше называлось планированием заданий будет выступать в качестве долгосрочного планирования. При планировании заданий решается какое очередное задание и стоящих в очереди на загрузку вычислительную систему должно быть в эту систему помещено.

Долгосрочное планирование определяет степень мультипрограммирования вычислительной системы то есть количество процессов, которые одновременно в ней находятся.

Краткосрочное планирование -- планирование процессов процессора. Часто используется в системах разделение времени

Планирование среднего уровня -- если в системе исполняется некоторое количество процессов, и в это время поступает задание первоочередной важности, можно взять некоторые частично выполнивший себе процесс и все содержимое памяти, контекст, сохранить временно на диск, освободив место для загрузки очень важного задания. После того, как задание будет выполнено вернем информацию с диска обратно в оперативную память и продолжим выполнять первый процесс процесс.

Цели планирования

Справедливость - процессы одного пользователя не имеют преимущества перед процессами другого пользователя

Эффективность использования вычислительной системы - процессор не простаивает

Минимизация полного времени нахождения процесса в системе - чем меньше это время, тем выгоднее пользователю

Сокращение среднего времени ожидания процесса или задание вычислительной системе - время нахождения в состоянии готовности

Сокращение времени отклика - время, которое проходит в вычислительной системе между нажатием клавиши на терминале и получением ответа на экране

Свойства алгоритмов планирования

Свойство предсказуемости поведения алгоритма - одна и та же задача на одном и том же в вычислительном комплексе должно решаться примерно за одно и то же время

Минимизация накладных расходов - сокращение времени работы самого алгоритма планирования

Равномерно загружать все ресурсы вычислительной системы - процессы, которые обращаются к малоиспользуемым ресурсам, должны получать преимущество перед другими процессами, чтобы ресурсы были загружены равномерно

Масштабируемость - при увеличении степени мультипрограммирования, работающий алгоритм не должен приводить к резкому ухудшению любого из критерий

Параметры планирования

1. Статические -- которые не меняются с течением времени (предельное значение ресурсов, объем дисковой памяти для смокинга количество устройств ввода-вывода). Эта информация должна быть известна еще до начала работы процесса, например, кто подготовил задание или запустил процесс, какой именно пользователь, каков приоритет данного задания или процесса, какие ресурсы запрашивают процесс для своего выполнения, сколько нужно оперативной памяти, какие и сколько устройств ввода-вывода и т.п
2. Динамические -- которые, наоборот, меняются. К динамическим параметрам системы мы можем отнести: текущее значение ресурсов, сколько сейчас в системе имеется свободной оперативной памяти, сколько имеется свободных устройств ввода-вывода и т.п

Деятельность может быть разбита на последовательность интервалов, когда процесс занимает процессор - реально исполняется и когда процесс вынужден дожидаться завершения операции ввода-вывода

CPU burst

Промежуток возможного непрерывного использования центрального процессора

Input-output burst

Промежуток времени, в течении которого мы занимаемся ожиданиям IO

Работа планировщика

Первая ситуация - когда у нас некоторый процесс завершил исполнение

Вторая ситуация - процесс, который исполнялся, обратился к операционной системе с просьбой произвести над ним блокирование, то есть он не может дальше выполняться

Третья ситуация - при осуществлении операций приостановка процесса или при переходе процесса из состояния ожидания

Невытесняющее планирование - когда CPU burst не истек полностью

Вытесняющее планирование - когда принудительно процесс был вытеснен с процессора и заменен другим процессом

Основные алгоритмы использования процессора

FCFS - First come - First Served

Первый пришел - первым обслужен (невытесняющего планирования)

По принципу FIFO

Процесс, который поступает в состоянии готовности, заносится в конец очереди готовых процессов

До тех пор пока, либо он не завершит исполнения, либо не попросит операционную систему заблокировать его

RR - Round Robin

Вытесняющая модификация алгоритма FCFS

Все процессы сидят на “карусели”, которая вращается. Рядом с каруселью находится процессор, и когда процесс, вращающийся на карусели, оказывается в окрестности процессора, исполняется, как только эту окрестность он покидает, он исполняться перестает и переходит состояние готовности

Каждая такая итерация происходит один раз в квант времени.

При завершении процесса до завершения кванта времени, то происходит переход к следующему процессу.

SJF - Shortest Job First

Невытесняющий вариант: после освобождения процессора процесс будет предоставлен в распоряжение процессу, имеющему наименьшее значение CPU Burst. При одинаковом значении - выбирается тот, что раньше пришел.

Вытесняющий алгоритм: не предполагает наличие квантов времени. Когда в очереди появляется новый процесс - смотрим, что больше -- CPU Burst нового процесса или остаток CPU Burst первого. Если у первого процесса он меньше, то вытеснение не происходит, если у нового -- вытеснение происходит.

SJF применяется только для долгосрочного планирования, потому что сложно предугадать CPU Burst каждого процесса.