Планирование и диспетчеризация процессов и задач

Стратегии планирования

Долгосрочное планирование заключается в подборе таких вычислительных процессов, которые бы меньше всего конкурировали между собой за ресурсы вычислительной системы. В современных ОС данное понятие не применяется.

При рассмотрении стратегий планирования говорят о краткосрочном планировании, т.е. диспетчеризации.

Стратегия планирования определяет, какие процессы мы планируем на выполнение для того, чтобы достичь поставленной цели. Существует большое количество различных стратегий выбора процесса, которому необходимо предоставить процессор. Среди них можно назвать следующие стратегии:

1) По возможности заканчивать вычисления (вычислительные процессы) в том же самом порядке, в котором они были начаты

2) Отдавать предпочтение более коротким процессам

3) Предоставлять всем пользователям (процессам пользователей) одинаковые услуги, в том числе и одинаковое время ожидания

Дисциплины диспетчеризации

При диспетчеризации в явном или неявном виде имеют ввиду понятие задачи (потока).

Известно большое количество правил (дисциплин диспетчеризации), в соответствии с которыми формируется список (очередь) готовых к выполнению задач.

Различают два больших класса дисциплин обслуживания — **бесприоритетные** и **приоритетные**.

При **бесприоритетном обслуживании** выбор задачи производится в некотором заранее установленном порядке без учета их относительной важности и времени обслуживания.

При реализации **приоритетных дисциплин обслуживания** отдельным задачам предоставляется преимущественное право попасть в состояние исполнения. При этом:

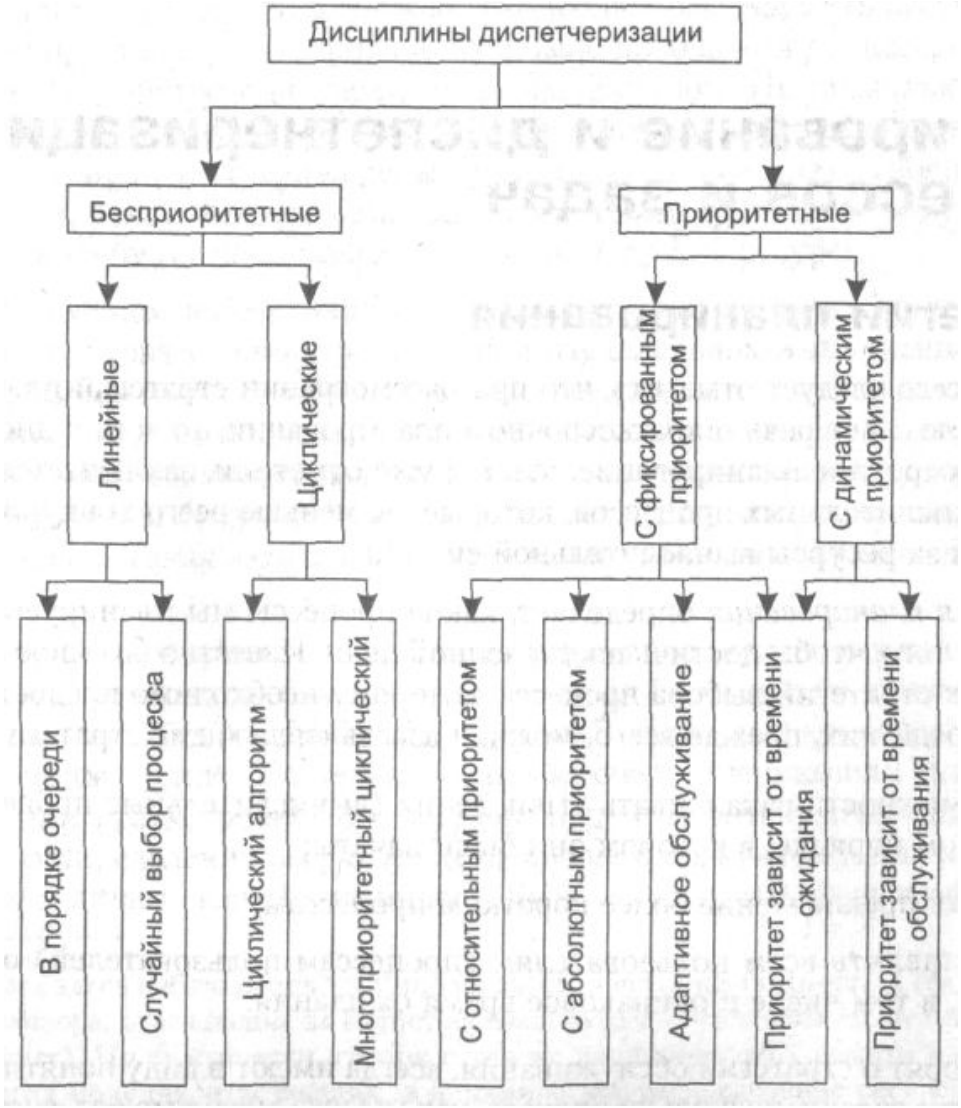
1) Приоритет, присвоенный задаче, может являться величиной постоянной
2) Приоритет задачи может изменяться (динамически) в процессе ее решения

Диспетчеризация с динамическими приоритетами требует дополнительных расходов на вычисление значений приоритетов исполняющихся задач. Однако динамические приоритеты позволяют реализовать гарантии обслуживания задач.

В большинстве ОСРВ используются методы диспетчеризации на основе статических (постоянных) приоритетов.

Дисциплины диспетчеризации

Классификация дисциплин диспетчеризации



Дисциплины диспетчеризации

Дисциплина FCFS (First come, first served; очередь) – является самой простой в реализации. В при ее использовании задачи обслуживаются «в порядке очереди», то есть в порядке их появления.

Особенность: те задачи, которые были заблокированы в процессе работы (попали в какое-либо из состояний ожидания, например, из-за операций ввода/вывода), после перехода в состояние готовности ставятся в эту очередь готовности перед теми задачами, которые еще не выполнялись. Т.е. образуются две очереди:

1) Очередь для новых задач
2) Очередь для ранее выполнявшихся задач

Дисциплины диспетчеризации

Дисциплина FCFS (First come, first served; очередь)

Такой подход позволяет реализовать стратегию обслуживания «по возможности заканчивать вычисления в порядке их появления». Эта дисциплина обслуживания не требует внешнего вмешательства в ход вычислений, при ней не происходит перераспределение процессорного времени.

Данная дисциплина диспетчеризации относится к классу не вытесняющих ДД.

Достоинства FCFS:

- Простота реализации
- Малые расходы системных ресурсов на формирование очереди задач

Недостатки FCFS:

- При увеличении загрузки вычислительной системы растет и среднее время ожидания обслуживания. «Короткие» задания (требующие небольших затрат машинного времени) вынуждены ожидать столько же, сколько и трудоемкие задания.

Избежать этого недостатка позволяют **дисциплины SJN** и **SRT**.

Дисциплины диспетчеризации

Дисциплина обслуживания SJN (shortest job next, следующим будет выполняться кратчайшее задание)

Она требует, чтобы для каждого задания была известна оценка в потребностях машинного времени.

Чтобы ОС «знала» о характеристиках задачи, в которых описывались потребности в ресурсах, были разработаны соответствующие языковые средства. Например, **язык JCL** (**job control language, язык управления заданиями**).

Пользователи вынуждены были указывать предполагаемое время выполнения. Чтобы избежать «злоупотреблениями» с их стороны (указывать заведомо меньшее время выполнения с целью получить результаты раньше других), ввели подсчет реальных потребностей. Диспетчер задач сравнивал заказанное время и время выполнения, и в случае превышения указанной оценки в данном ресурсе ставил данное задание не в начало, а в конец очереди. В некоторых ОС в таких случаях использовалась система штрафов, при которой в случае превышения заказанного машинного времени «оплата» вычислительных ресурсов осуществлялась уже по другим «расценкам».

Дисциплина обслуживания SJN предполагает, что *имеется только одна очередь заданий, готовых к выполнению*. Задания, которые в процессе своего исполнения были временно заблокированы (например, ожидали завершения операций ввода/вывода), вновь попадают в конец очереди готовых к выполнению наравне с вновь поступающими. Это приводит к тому, что задания, которым требуется очень немного времени для своего завершения, вынуждены ожидать процессор наравне с длительными работами, что не всегда хорошо.

Дисциплины диспетчеризации

Дисциплина обслуживания SRT (shortest remaining time, следующее задание требует меньше всего времени для своего завершения)

В отличии от предыдущей SJN, данная дисциплина обслуживания предполагает, что в начало очереди будут ставиться те задачи, которым для своего завершения осталось меньше всего времени (не важно, «короткие» это задачи или «длинные»). Таким образом, с точки зрения очереди, задачи будут завершаться быстрее.

Все эти три дисциплины обслуживания (FCFS, SJN, SRT) могут использоваться для пакетных режимов обработки, когда пользователь не вынужден ожидать реакции системы, а просто сдает свое задание и потом получает свои результаты вычислений.

Для интерактивных вычислений желательно прежде всего обеспечить приемлемое время реакции системы и равенство в обслуживании, если система является мультитерминальной. Если же это однопользовательская система, но с возможностью мультипрограммной обработки, то желательно, чтобы те программы, с которыми мы сейчас непосредственно работаем, имели лучшее время реакции, нежели наши фоновые задания.

При этом некоторые приложения могут выполняться без нашего непосредственного участия (например, программа получения электронной почты), тем не менее гарантированно получали необходимую им долю процессорного времени.

Для решения подобных проблем используется дисциплина обслуживания, называемая **RR (round robin круговая, карусельная), и приоритетные методы обслуживания**.

Дисциплины диспетчеризации

Дисциплина обслуживания RR (round robin, круговая, карусельная)

Дисциплина обслуживания RR предполагает, что каждая задача получает процессорное время порциями (выделяется квант времени q). После окончания кванта времени q задача снимается с процессора и он передается следующей задаче. Снятая задача ставится в конец очереди задач, готовых к выполнению.

Для оптимальной работы системы необходимо правильно выбрать закон, по которому кванты времени выделяются задачам. Величина кванта времени q выбирается как компромисс между приемлемым временем реакции системы на запросы пользователей (с тем, чтобы их простейшие запросы не вызывали длительного ожидания) и накладными расходами на частую смену контекста задач.

Дисциплины диспетчеризации

Дисциплина обслуживания RR (round robin, круговая, карусельная)

При прерываниях ОС вынуждена сохранить достаточно большой объем информации о текущем (прерываемом) процессе, поставить дескриптор снятой задачи в очередь, загрузить контекст задачи, которая теперь будет выполняться (ее дескриптор был первым в очереди готовых к исполнению).

Проблема «большого» кванта времени: Если величина q велика, то при увеличении очереди готовых к выполнению задач реакция системы станет плохой.

Проблема «малого» кванта времени: Если же величина q мала, то относительная доля накладных расходов на переключения между исполняющимися задачами станет большой и это ухудшит производительность системы.

Примечание: в некоторых ОС есть возможность указывать в явном виде величину q либо диапазон ее возможных значений, поскольку система будет стараться выбирать оптимальное значение сама.

Дисциплины диспетчеризации

Дисциплина обслуживания RR (round robin, круговая, карусельная)

Дисциплина диспетчеризации RR — это одна из самых распространенных дисциплин. Однако бывают ситуации, когда ОС не поддерживает в явном виде дисциплину карусельной диспетчеризации. Например, в некоторых ОСРВ используется диспетчер задач, работающий по принципам абсолютных приоритетов (вперед выполняются задачи с максимальным приоритетом) и тогда нужно самостоятельно создать подобную дисциплину диспетчеризации. Для этого достаточно всем пользовательским задачам присвоить одинаковые приоритеты и создать одну высокоприоритетную задачу, которая не должна ничего делать, но которая, тем не менее, будет по таймеру (через указанные интервалы времени) планироваться на выполнение. Эта задача снимет с выполнения текущее приложение, оно будет поставлено в конец очереди, и поскольку этой высокоприоритетной задаче на самом деле ничего делать не надо, то она тут же освободит процессор и из очереди готовности будет взята следующая задача. В своей простейшей реализации дисциплина карусельной диспетчеризации предполагает, что все задачи имеют одинаковый приоритет.

Если же необходимо ввести механизм приоритетного обслуживания, то это, как правило, делается за счет организации нескольких очередей. Процессорное время будет предоставляться в первую очередь тем задачам, которые стоят в самой привилегированной очереди. Если она пустая, то диспетчер задач начнет просматривать остальные очереди. По такому алгоритму действует диспетчер задач в операционных системах OS/2 и Windows NT (и современные ОС Windows).