

- Цель:

Обеспечение работы задач, процессов

- Задачи:
  - а. Максимальную производительность и минимальное время выполнения для всех процессов
  - б. Гарантировать соблюдение приоритета процесса
  - с. Многопроцессорных должны определять не только порядок выполнения процессов, но и то, на каких процессорах они должны выполняться
- Отличие задачно-независимых от задачно-ориентированных алгоритмов

Задачей-независимые требуют для своей реализации минимум накладных расходов, так как они не пытаются добиться максимального распараллеливания или воспользоваться привязкой задачи к процессору. Задачно-ориентированные оценивают свойство каждой задачи и пытаются достичь максимального ее распараллеливания или мягкой привязки к процессору.

- Обеспечение равномерного распределения за счёт чего обеспечивается

Глобальная очередь

- Как изменить задачно-независимый алгоритм, чтобы точно получился задачно-ориентированный.

Надо сгруппировать задачи в глобальной очереди

- Чем отличается гг ЗН от гг ЗО. И что случится если процесс задачи уйдёт на ввод/вывод в гг ЗО.

Надо рассказать про очереди процессов и сказать, что если на ввод-вывод, то общий квант распределяется между оставшимися равномерно

## **Алгоритмы планирования**

У планировщика в многопроцессорных системах те же самые задачи, что и в однопроцессорных. Планировщик должен обеспечивать максимальную производительность и минимальное время выполнения для всех процессов. Кроме того система должна гарантировать соблюдение приоритета процесса. В отличие от алгоритмов планирования, используемых в однопроцессорных системах алгоритмы для многопроцессорных должны определять не только порядок выполнения процессов, но и то, на каких процессорах они должны выполняться. За счёт этого растёт сложность алгоритмов. Например, алгоритмы планирования для многопроцессорных систем должны гарантировать, что процессоры в системе не простаивают, если есть процессы ожидающие выполнения. Когда нужно определить на каком процессоре должен выполняться процесс планировщик принимает во внимание несколько моментов. Например, в некоторых стратегиях планирования планировщик старается максимально распараллелить работы и полностью использовать возможности системы по одновременному выполнению программ. Взаимодействующие процессы в системах часто объединяются в задачи. Параллельное выполнение процессов одной задачи позволяет

повысить производительность поскольку процессы выполняются действительно одновременно. Существует несколько алгоритмов планирования с временным разделением, пытающихся воспользоваться такой параллельностью по планировании размещения взаимодействующих процессов на разные процессоры. Это позволяет процессам более эффективно синхронизировать свою одновременную работу.

Другие стратегии сосредотачиваются на привязке задач процессора, т.е. взаимосвязи процесса и процессора к его локальной памяти и кэша.

Процесс сильно привязанный к процессору выполняется на этом процессоре в течение большей части своего жизненного цикла. Преимущества такого подхода заключается в том, что процесс будет чаще получать требуемые данные из кэша этого процессора. ... . Алгоритмы планирования, пытающиеся разместить по возможности процесс на один процессор на все время выполнения этого процесса, обеспечивают мягкую привязку. А алгоритмы размещающие процесс всегда на одном и том же процессоре обеспечивают жесткую привязку. Алгоритмы планирования с пространственным разделением пытаются добиться максимальной привязки задач к процессорам, направляя задачи для выполнения на один процессор или группу тесно связанных процессоров. Эти алгоритмы подразумевают, что взаимодействующие процессы будут работать с одними и теми же данными, и эти данные скорее всего удастся разместить в кэшах процессоров и их локальной памяти. Поэтому алгоритмы планирования с пространственным разделением увеличивают частоту успешных обращений к кэшу и локальной памяти. Однако, они могут ограничить производительность поскольку при использовании взаимодействующие процессы обычно не исполняются одновременно.

Алгоритмы планирования в многопроцессорных системах делятся на заданной-ориентированные и задачно-независимые. Задачно-независимые требуют для своей реализации минимум накладных расходов, так как они не пытаются добиться максимального распараллеливания или воспользоваться привязкой задачи к процессору. Задачно-ориентированные оценивают свойство каждой задачи и пытаются достичь максимального ее распараллеливания или мягкой привязки к процессору. Таким образом повышая производительность за счёт дополнительных накладных расходов.

Во многих алгоритмах планирования для многопроцессорных систем, процессы организуются в глобальную очередь выполнения. Каждая такая очередь содержит все процессы в системе, которые готовы к выполнению.

Глобальные очереди выполнения можно использовать для сортировки процессов по их приоритетам, по задачам, к которым они относятся и др. В системах могут также использоваться процессоров очереди выполнения. Использование таких очередей характерно для больших слабосвязанных систем, в которых нужно добиться как можно большего уровня успешных обращений к кэшу и локальной памяти. В процессорных очередях выполнения процессы ассоциируются с определённым процессором и система реализует определенный алгоритм планирования для каждого процессора. В некоторых системах используются узловые очереди выполнения. В таких системах в каждом узле может находиться по несколько процессоров. Эти очереди эффективны в системах, в которых процессы связаны с определенными группами процессоров. С проблемой планирования. Связана и проблема миграции процессов. Т.е. перенос процессов из одной процессоров или узловой очереди в другую.

- FCFS

Есть общая очередь процессы загружаются на процессор, закончил освободил, пошёл следующей. Нет привязки к задачам. Не используется квантование. Процесс выполняется сколько хочет.

+

Процесс выполняется в порядке поступления

Нет бесконечного откладывания, т.к. нет приоритетов

Равномерная загрузка процессоров

Простота реализации

-

Более важные процессы придётся ждать

Алгоритм не пригоден для планирования интерактивных процессов, т.к. должны быть равные условия, а надо квантованием

Не обеспечивает высокой степени параллельности

Нет привязки к процессорам, глобальная очередь. Связи между процессорами нет.

- SPF

Запускается процесс располнел время выполнение которого требуется меньшее время.

+

Меньше времени ожидания для интерактивных процессов

Есть квантование

-

Бесконечное откладывание

Нет параллелизма и привязки к процессорам