

Раздел 3. МАШИННО-НЕЗАВИСИМЫЕ СВОЙСТВА ОПЕРАЦИОННЫХ СИСТЕМ

Свойства, которыми обладают операционные системы, делятся на две группы: **машинно-зависимые** и **машинно-независимые**.

Машинно-зависимые свойства современных ОС, такие как: *прерывания, планирование процессов, управление реальной и виртуальной памятью, обслуживание ввода-вывода*, - были рассмотрены в предыдущих лекциях.

К **машинно-независимым** свойствам операционных систем относятся:

- работа с файлами;
- управление заданиями;
- распределение ресурсов;
- защита данных.

Тема 3.2. ПЛАНИРОВАНИЕ ЗАДАНИЙ. РАСПРЕДЕЛЕНИЕ РЕСУРСОВ

Тема 3.2.2 Распределение ресурсов. Взаимоблокировки.
Обнаружение и устранение взаимоблокировок

План занятия:

1. Распределение ресурсов.
2. Взаимоблокировки.
3. Обнаружение и устранение взаимоблокировок.
4. Предотвращение взаимоблокировок.

1. РАСПРЕДЕЛЕНИЕ РЕСУРСОВ

Часто при выполнении прикладных задач возникают ситуации, когда *один процесс* нуждается в доступе сразу к *нескольким ресурсам*, или *один ресурс* необходим сразу *нескольким процессам*, и они начинают бороться за него.

Ресурсы – объекты предоставления доступа.

Ресурсы бывают *выгружаемые* и *невыгружаемые*, *аппаратные* и *программные*.

Выгружаемый ресурс - этот ресурс безболезненно можно забрать у процесса (*например*: память).

Невыгружаемый ресурс - этот ресурс нельзя забрать у процесса без потери данных (*например*: принтер).

Последовательность событий, необходимых для использования ресурса, можно представить в *абстрактной* форме таким образом:

- Запрос ресурса.
- Использование ресурса.
- Возврат ресурса.

2. ВЗАИМОБЛОКИРОВКИ

Когда несколько процессов *борются* за один ресурс, может произойти *взаимоблокировка* процессов. Такая проблема возникает при борьбе за невыгружаемые ресурсы (*например*: принтер).

Группа процессов находится в **тупиковой ситуации (взаимоблокировке)**, если каждый процесс из группы ожидает событие, которое может вызвать только другой процесс из той же группы.

Например, в системе исполняются два процесса, каждый из которых хочет записать отсканированный документ на компакт-диск. Процесс *A* запрашивает разрешение на исполнение сканера и получает его. Процесс *B* запрашивает устройство для записи компакт-дисков и получает его. Затем процесс *A* обращается к устройству для записи компакт-дисков, но запрос *отклоняется* до тех пор, пока то устройство занято процессом *B*. Вместо того, чтобы *освободить* устройство для записи компакт-дисков, процесс *B* запрашивает сканер. В итоге процессы *блокируются*. Такая ситуация называется *тупиком*, тупиковой ситуацией, или *взаимоблокировкой*.

Для возникновения ситуации взаимоблокировки должны выполняться **четыре** условия:

- 1. **Условие взаимного исключения.** Каждый ресурс в данный момент или отдан ровно одному процессу, или доступен.
- 2. **Условие удержания и ожидания.** Процессы, в данный момент удерживающие полученные ранее ресурсы, могут запрашивать новые ресурсы.
- 3. **Условие отсутствия принудительной выгрузки ресурса.** У процесса нельзя принудительным образом забрать ранее полученные ресурсы. Процесс, владеющий ими, должен сам освободить ресурсы.
- 4. **Условие циклического ожидания.** Должна существовать круговая последовательность из двух и более процессов, каждый из которых ждет доступа к ресурсу, удерживаемому следующим членом последовательности.

3. ОБНАРУЖЕНИЕ И УСТРАНЕНИЕ ВЗАИМОБЛОКИРОВОК.

Алгоритмы обнаружения взаимоблокировок с целью их последующего устранения можно представить в виде моделей (графов).

Графы имеют два вида узлов:

- **процессы**, показанные кружочками,
- **ресурсы**, нарисованные квадратиками.

Ребро, направленное от узла **ресурса** (квадрат) к узлу **процесса** (круг), означает, что *ресурс* ранее был **запрошен процессом**, получен и в данный момент используется этим процессом.

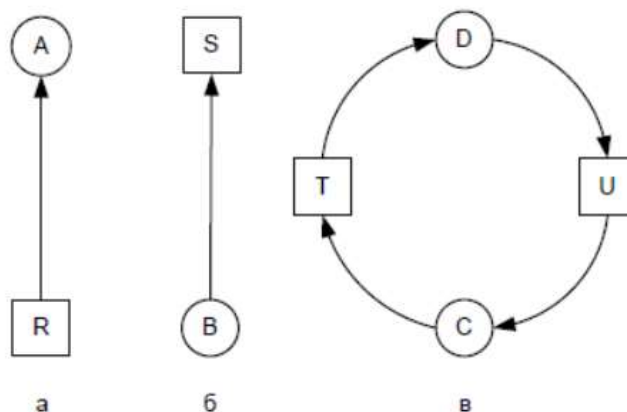


Рис. 1. Графы распределения ресурсов: ресурс занят (а); запрос ресурса (б); взаимоблокировка (в)

На Рис. 1(а) ресурс **R** в настоящее время отдан процессу **A**.

Ребро, направленное от процесса к ресурсу, означает, что процесс в данный момент *блокирован* и находится в состоянии *ожидания* доступа к этому ресурсу.

Для избежания взаимоблокировок существуют следующие стратегии:

- **пренебрежение проблемой в целом** (возможно взаимоблокировка не возникнет);
- **обнаружение и устранение** (взаимоблокировка происходит, но оперативно ликвидируется);
- **динамическое избежание взаимоблокировок;**
- **предотвращение условий возникновения взаимоблокировок.**

В первом случае, если вероятность взаимоблокировки очень мала, то ею легче пренебречь, т.к. код исключения может очень усложнить ОС и привести к большим ошибкам. Также многие взаимоблокировки тяжело обнаружить. Этот алгоритм используется как в UNIX,

так и в Windows. Поэтому (и не только) на серверах часто устанавливают *автоматическую перезагрузку* (раз в сутки, как правило, ночью), если возникнет взаимоблокировка, то после перезагрузки ее не будет.

Во втором случае система не пытается предотвратить взаимоблокировку, а пытается обнаружить ее и устранить. Обнаружить взаимоблокировку можно:

- когда запрашивается очередной ресурс;
- через какой-то промежуток времени;
- когда загрузка процессора слишком велика.

После обнаружения взаимоблокировки возможны следующие способы выхода из нее:

1. при помощи принудительной выгрузки ресурса (как правило, требуется ручное вмешательство);

2. через откат (состояние процессов записывается в контрольных точках, и в случае тупика можно сделать откат процесса, после чего он продолжит работу снова с этой точки);

3. путем уничтожения процесса.

В третьем случае ОС должна знать наперед, является ли предоставление ресурса безопасным или нет.

4. ПРЕДОТВРАЩЕНИЕ ВЗАИМОБЛОКИРОВОК

Для возникновения взаимоблокировок необходим ряд условий. Предотвращение этих условий предотвратит и сами взаимоблокировки.

Предотвращение условия взаимного исключения состоит в минимизации количества процессов, борющихся за ресурсы.

Предотвращение условия удержания и ожидания состоит в том, что процесс должен запрашивать все необходимые ресурсы до начала работы. Если хоть один ресурс недоступен, то процессу вообще ничего не предоставляется

Предотвращение условия отсутствия принудительной выгрузки ресурса состоит в том, что система разрешает выгружать ресурсы.

Предотвращение условия циклического ожидания осуществляется следующими способами:

- прежде чем занять новый ресурс, процесс освобождает старый;
- упорядочение (нумерация) ресурсов, в результате чего процессы запрашивают ресурсы только по порядку.

Вопросы к теме "Распределение ресурсов"

1. Что такое взаимоблокировка?
2. При работе, с какими ресурсами может возникнуть взаимоблокировка?
3. Что такое ресурс?
4. Что может быть ресурсом?
5. Какой алгоритм использует система для обнаружения и устранения взаимоблокировок?
6. Как можно избежать взаимоблокировок?
7. Перечислите 4 основных условий, которые приводят к взаимоблокировке.
8. Дайте определение термину "подкачка"
9. Дайте определение термину "демон"
10. При каких условиях предотвращение взаимоблокировки не возможно?