

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ»

Г.Э. Вошинская, М.А. Артемов

ОПЕРАЦИОННЫЕ СИСТЕМЫ

Часть 2

Учебно-методическое пособие для вузов

Издательско-полиграфический центр
Воронежского государственного университета
2012

Содержание

ТУПИКИ	5
Условия наличия тупика	5
Предотвращение тупиков.....	5
Обход тупиков.....	6
Обнаружение тупиков	6
Восстановление после тупиков	6
Алгоритмы предотвращения тупиков.....	6
Выделение всех необходимых ресурсов	6
Выделение ресурсов в порядке присвоенных номеров	6
Метод Габермана	7
Алгоритм банкира.....	8
Тупики как критический фактор для будущих систем	10
Управление памятью	10
Типы адресов	11
Методы распределения памяти без использования дискового пространства	12
Распределение памяти фиксированными разделами	12
Распределение памяти разделами переменной величины.....	13
Перемещаемые разделы	14
Понятие виртуальной памяти	14
Страничное распределение	15
Сегментное распределение	18
Странично-сегментное распределение	19
Свопинг	19
Иерархия запоминающих устройств. Принцип кэширования данных ...	20
Поиск файлов.....	22
Запись TSearchRec	22
Атрибуты файла	23
Перебор файлов с использованием C#.	26
ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОГО ВЫПОЛНЕНИЯ	30
Моментальные снимки в Windows 95/98: использование ToolHelp32.....	33
Обработка информации о процессах	34
Обработка информации о потоках	36
Обработка информации о модулях	37
Обработка информации о динамической памяти (кучах)	39

когда какой-нибудь процесс делает запрос, который может привести к тупику, система принимает меры к тому, чтобы избежать опасного состояния: либо не удовлетворяет этот запрос, либо отбирает ресурс у другого процесса, чтобы избежать возможного попадания в тупик. Достоинство такого подхода в полном исключении тупиков. Недостаток: такой подход часто приводит к нерациональному использованию ресурсов, да и сам предотвращающий алгоритм может внести большие накладные расходы.

Обход тупиков

Цель средств обхода тупиков заключается в том, чтобы можно было предусмотреть менее жесткие ограничения, чем в случае предотвращения тупиков. Методы обхода тупиков учитывают возможность возникновения тупика, однако в этом случае принимаются меры по аккуратному обходу тупика.

Обнаружение тупиков

Методы обнаружения тупиков применяются в системах, допускающих возможность возникновения тупиков. Когда это происходит, система обнаруживает тупик программным путём и принимает меры для вывода из тупика (например, перераспределяя ресурсы). Выход из тупика может выполняться автоматически или под управлением оператора.

Восстановление после тупиков

Методы восстановления после тупиков применяются для устранения тупиковых ситуаций. Например:

- восстановление при помощи принудительной выгрузки;
- восстановление через откат;
- восстановление путем уничтожения одного или нескольких процессов.

Алгоритмы предотвращения тупиков

Рассмотрим некоторые алгоритмы предотвращения тупиков.

Выделение всех необходимых ресурсов

Процесс получает все необходимые ему ресурсы перед началом работы. В этом случае возникновение тупика исключается, но нет разделения ресурсов и использование ресурсов нерационально.

Выделение ресурсов в порядке присвоенных номеров

Все разделяемые ресурсы в системе пронумерованы. Процесс запрашивает ресурсы по возрастанию номеров. Он не может запрашивать следующий ресурс, если предыдущий запрос не удовлетворен. Например, процесс использует сначала ресурс 3, затем ресурс 5, затем ресурс 2. Тогда, чтобы получить ресурс 3, он должен сначала запросить и получить ресурс 2, а затем запрашивать ресурс 3. Ресурс 5 он может получить позже, когда он

ему понадобится. Этот алгоритм допускает разделение ресурсов, условия выделения ресурсов менее жесткие, чем в предыдущем способе, но, в некоторых случаях, когда порядок работы процесса с ресурсами не совпадает с нумерацией ресурсов, процесс должен захватывать и удерживать ресурс заранее, хотя использовать его он будет позже.

Метод Габермана

Системе требуются

а) ориентированный граф, в котором узлы соответствуют процессам, а дуга проводится от узла i к узлу j , если процесс j может запросить ресурс, который запрашивает процесс i ;

б) предварительная информация о ресурсах, необходимых каждому процессу. Она хранится в таблице, в которой строки соответствуют процессам, а столбцы – ресурсам;

с) таблица учета выделенных ресурсов.

Правило Габермана гласит: состояние является опасным, если граф содержит цикл.

Алгоритм вызывается при каждом запросе и возврате ресурса.

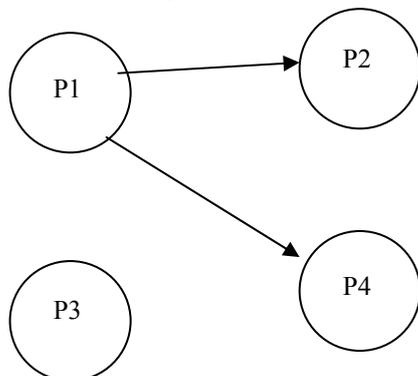
Например.

Пусть в системе выполняются процессы P1, P2, P3, P4, и ими используются ресурсы A, B, C, D. В таблице для каждого процесса 1 отмечены ресурсы, которые ему могут понадобиться.

	A	B	C	D
P1	1	1		
P2	1	1	1	
P3		1	1	1
P4	1		1	1

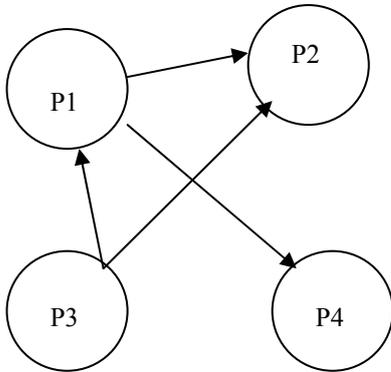
Предположим следующий сценарий запросов.

Пусть P1 запрашивает A. Ресурс свободен. Ресурс A могут еще запросить процессы P2 и P4. В графе проводятся дуги от P1 к P2 и от P1 к P4. Цикла нет – ресурс выделяется.



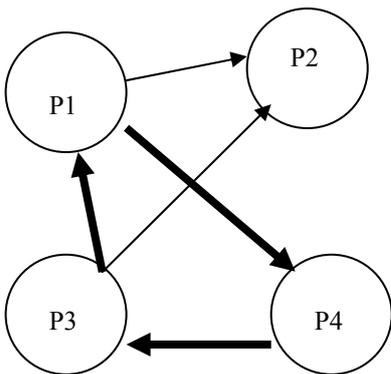
Пусть P2 запрашивает А. Ресурс занят – процесс P2 блокируется.

Пусть P3 запрашивает В. Ресурс свободен. Но ресурс В могут запросить также процессы P1 и P2. В графе проводятся дуги от P3 к P1 и от P3 к P2.



Цикла нет – ресурс выделяется.

Пусть P4 запрашивает D. Ресурс свободен. Но ресурс D может запросить процесс P3. В графе проводится дуга от P4 к P3.



Есть цикл – ресурс не выделяется. Процесс P4 блокируется. Граф возвращается в предыдущее состояние.

Алгоритм банкира

Этот алгоритм используется для распределения делимых ресурсов. Рассмотрим его идею на примере. Пусть в системе имеется 10 единиц некоторого ресурса и работают 3 процесса, использующих этот ресурс. В таблице содержится информация о максимальной потребности каждого процесса и о количестве уже выделенных единиц ресурса. Предполагается, что процессы используют необходимые ресурсы не обязательно сразу, а могут запрашивать их по частям.

процесс	max потр.	выделено
P1	4	2
P2	7	3
P3	8	2