Лекция 04

РИС, III курс, ИСиТ

**Алгоритмы взаимного исключения**

1. **Напоминание.**



1. **Постановка задачи**: несколько распределенных процессов дописывают данные в конец общего файла на файловом сервере.



1. **Механизмы синхронизации**: критическая секция – механизм синхронизации (взаимного исключение) потоков в рамках одного процесса.
2. **Алгоритмы взаимного исключения:** централизованный алгоритм.

Один процесс, работающий с распределенным ресурсом



Три процесса, работающих с распределенным ресурсом









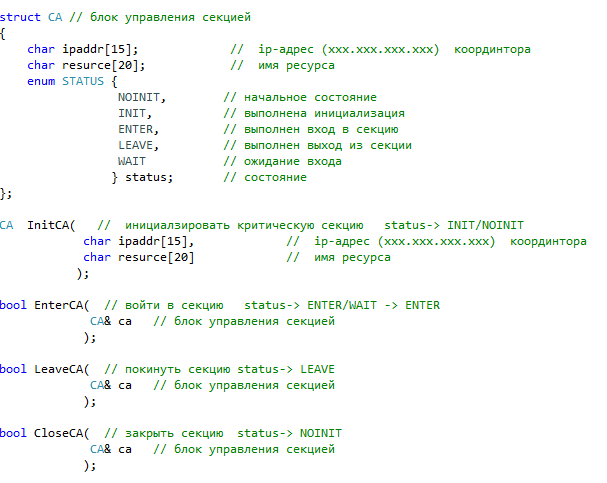


1. **Алгоритмы взаимного исключения:** уровень доступа к ресурсу (файлы: open/close, write record).
2. **Алгоритмы взаимного исключения:** распределенный алгоритм. Надежная связь между узлами. Каждый узел, должен знать о других узлах участвующих в синхронизации.

**Принцип:**

* процесс не собирается входить: на все запросы **enter(метка времени)** отвечает **OK**;
* процесс собирается войти в критическую область: рассылает всем запрос **enter(метка времени)** и ждет от всех **OK**; если получил от всех **OK** выполняет действие;
* процесс собирается войти в критическую область (разослал **enter(метка времени)**) и одновременно получает **enter(метка времени)** от другого процесса: сравнивает свою метку времени и чужую; если чужая метка с меньшим временем – отсылает **OК**  и ждет **OK** от всех других; если собственное время меньше, то чужой **enter(метка времени)** поместить в очередь, после выхода всем тем, кто в очереди – оправляется **OK** и соответствующие **enter(метка времени)** извлекаются из очереди;
* процесс вошел в критическую область: поступает **enter(метка времени)** – помещается в очередь; после выхода всем тем, кто в очереди – оправляется **OK** и соответствующие **enter(метка времени)** извлекаются из очереди.

1. **Алгоритмы взаимного исключения:** кольцевоймаркер. Узлы связываются в односвязный кольцевой список. По списку циркулирует маркер, предлагающий войти в критическую область; если входить в критическую область не требуется, то маркер отправляется дальше; если необходимо войти, то выполняется критическая операция и после выполнения маркер отправляется дальше.

1. **Сравнение:** общая проблема – разрыв связи; централизованный алгоритм – сбой координатора; распределенный алгоритм – сбой любого процесса; маркер – потеря маркера (сбой в одном из процессов).
2. **Лабораторная работа**
3. ****