Средства взаимодействия параллельных процессов

На лекции рассматривались семафоры Дейкстры. Семафор Дейкстры был определен как защещенная положительная переменная, на которой определены две операции, которые он определил как P(S) и V(S).

P - захват семафора (декремент). Возможна, если значение семафора больше 0. Если опреацию выполнит ь нельзя, то процесс блокируется на семафоре.

V - освобождение семафора (инкремент).

P и V не являются эквивалентными понятиями к lock и unlock.

Сама идея семафоров родилась на основе суммирования проблем взаимодействия процессов. Особенно - проблема активного ожидания на процессоре. Плата за это - переход в режим ядра (переключение аппаратного контекста минимум два раза) (только система может заблокировать и разблокировать процесс).

Для структурирований действия с семафорами, системы поддерживают наборы считающих семафоров. Примеры - на основе задачи об обедающих философов.

У семафора нет хозяина: семафор может освободить любой процесс.

В ядре UNIX/Linux имеется таблица семафоров.

[картинка]

В этой таблице содержатся дескрипторы наборов семафоров.

Библиотека - <sys/sem.h>.

sem_base - указатель на набор.

Информация:

- Идентификатор - целое число. Присваивается процессом, который создал набор. Другие процессы по этому идентификатору могут получить дескриптор набора и оперировать этими наборами (!!!).
- UID создателя набора семафоров. Процесс, эффективный UID которого совпадает с UID создателя, может удалять и менять(???) набор.
- Права доступа (user, group, others)
- Количество семафоров в наборе
- Время изменения одного или нескольких семафоров
- Время изменения параметров набора(???)
- Указатель на массив семафоров

О каждом семафоре известно:

[...]

На семафоре выделены следующие системные вызовы:

semget() - создание набора семафоров

semctl() - изменение параметров семафоров

semop() - операция на семафоре

[картинка с комментариями]

В System V определено три операции.

Операции sem_op = 0 нет у Дейкстры (если семафор захвачен - блокируется процесс; без захвата).

sem_op > 0 - инкремент.

На семафорах определены специальные флаги:

- IPC_NOWAIT - [...]. Сделано для того, чтобы избежать блокировки всех процессов, находящихся в очереди для доступа к ресурсам, если захвативший ресурс процесс завершился аварийно или получил сигнал kill. Поскольку kill нельзя перехватить, то процесс не может освободить семафор, и все процессы в очереди буду заблоикрованы.

- SEM_UNDO - указывает ядру на необходимость отслеживать [...]. При завершении процесса ядро ликвидирует сделанные процессом изменения. Добавлен по той же причине, что и IPC_NOWAIT.

[пример]

В данном примере создается набор с идентификатором 100 и количеством семафоров 2. Если удалось создать набор, вызывается sem_op, в который передается массив структур.

Данный пример также демонстрирует, что процесс не обязан освобождать уже захваченный семафор.

Сегменты разделяемой памяти

Это средство передачи информации между процессами.

[сопоставление программных каналов и сегментов разделяемой памяти]

[таблица сегментов разделяемой памяти]

shmget()

shmctl()

shmat()

shmdt()

На сегментах не определены методы взаимоисключения. Просто участок памяти. Просто пишем и читаем. Но для того, чтобы обеспечивать монопольный доступ к разделяемой памяти, используются в семафоре.