Понятие о POSIX семафорах

В стандарте POSIX вводятся другие семафоры, полностью аналогичные семафорам Дейкстры. Для инициализации значения таких семафоров применяется функция *sem_init()*, аналогом операции **P** служит функция *sem_wait()*, а аналогом операции **V** - функция *sem_post()*. К сожалению, во многих версиях Linux такие семафоры реализованы только для нитей исполнения внутри одного процесса, и поэтому подробно мы на них останавливаться не будем. Вместо этого мы рассмотрим мьютексы на семинаре 13.

На семинаре 5, когда вы изучали связь родственных процессов через pipe, мы говорили о том, что pipe является однонаправленным каналом связи, и что для организации связи через один pipe в двух направлениях необходимо использовать механизмы взаимной синхронизации процессов.

Задача на семинар

***Задача 3 (15 баллов)**:*

*Взяв за основу программу **05-3.c**, реализуйте поочередную неоднократную передачу информации между ребёнком и родителем через один pipe в двух направлениях, используя ровно один IPC семафор. Считать, что сначала родитель передает данные ребёнку, а затем получает данные от него. Количество пар передача и приём данных в родителе и пар приём и передача в ребёнке равно **N ≥ 2**. Значение **N** вводится в родителе с клавиатуры до порождения ребёнка. Число **N** потенциально может быть достаточно большим и превышать максимально возможное значение семафора, поэтому использовать значение **N** в операциях над семафором не разрешается.*

***Примечание**:*

*Основная сложность в решении задачи заключается в организации правильной логики работы с семафором. Поэтому вы **обязаны** написать в начале программы комментарий с описанием логики работы в терминах операций **A**, **D**, **Z**. В этом комментарии должно быть указано начальное значение семафора, взаимное расположение операций над семафорами и вызовы write(), read() в циклах приёма-передачи для родителя и ребёнка.*

Если вы выполняете задачу на семинаре, то перед тем, как начать реализовывать программу, имеет смысл показать алгоритм преподавателю.

Домашнее задание

Дедлайн для вашей группы прописывается в таблице с результатами. Сдать задание нужно до наступления указанного дня, сдавать в указанную дату уже поздно.

Проверка производится один раз, штрафы за недочеты выставляются сразу, но до дедлайна вы можете попросить проверить ваш алгоритм.

***Задача 1 (20 баллов)**:*

Рассмотрим катер, совершающий кольцевые туристические рейсы по реке Москве. Катер с утра приплывает на пристань и совершает ровно K поездок. В плавание он не отправляется, пока не будут заполнены все места, вместимость катера равна N (число N относительно невелико). Считаем, что пассажиры совершают поездку по одиночке, а не в компании. Пассажир приходит на пристань, ждет прибытия катера и выхода прибывших пассажиров, и после этого в порядке очереди заходит на катер. Трап узкий – поэтому все пассажиры заходят друг за другом. Когда все места заполнены — выполняется путешествие, после прибытия обратно пассажиры выгружаются строго по одному. Повторные поездки пассажиры не совершают.

*Смоделируйте поведение катера и каждого из пассажиров отдельными процессами, синхронизировав их действия с помощью семафоров System V IPC. Для этого напишите две программы – одна из них моделирует поведение катера, совершающего K поездок, другая генерирует N*K идентичных процессов, моделирующих поведение пассажиров. И катер, и пассажиры должны детально докладывать о своих действиях. Считать при тестировании, что N=5, K=3. Количество семафоров не должно превышать значения 3.*

*К решению **должно** прилагаться описание алгоритма, аналогичное задаче 10.1. Данное решение **можно** обсудить с преподавателем до дедлайна.*

В GitLab CI считается, что числа N и K передаются на вход программе-катеру и программе-генератору пассажиров с клавиатуры в порядке N и K. Вы вполне можете не следовать этой логике, если не пользуетесь CI.