Отчёт по ИДЗ №2

Кормилицын Владимир Алексеевич, БПИ 226

Вариант 28

Отчёт о выполнении:

Условие задания:

Изображение выглядит как текст, снимок экрана, Шрифт, письмо

Автоматически созданное описание

Программа на 4-5 баллов:

Для решения задачи создаётся из входных аргументов командной строки получаются количества процессов для первых, вторых и третьих рабочих (т.е. рабочих на 1, 2 и 3 участках)

Далее при помощи функции init\_resources инициализируются ресурсы приложения (структура описана далее) и создаются новые процессы в функции run\_program. В конце, родительский процесс, в котором запущена функция main, высвобождает ресурсы при помощи функции deinit\_resources.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Структура Resources содержит 2 структуры SharedData – для связи первых рабочих со втормым и для связи вторых рабочих с третьими.

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Структура SharedData – пара из файлового дескриптора общей памяти и указателя на общую память SharedMemory, представляющую собой циклическую очередь.

Для объекта “булавка” была создана структура Pin, хранящая id булавки.

Ресурсы корректно высвобождаются, т.к. в процессах-потомках выход происходит при помощи функции exit, и они не доходят до вызова функции deinit\_resources, а в основном процессе игнорируется сигнал SIGINT: когда пользователь нажимает ctr+c, процессы-потомки завершают работу (для них для сигнала SIGINT установлен коллбек worker\_shutdown\_signal\_handler),

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание  
и при этом родительский процесс ждёт завершения работы всех потомков, высобождая после этого все ресурсы.

Функции для управления ресурсами приложения объявлены в resource.h и определены в resource.c.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Для имён разделяемой Posix памяти и именованных семафороф Posix используются константы, объявленные в memory\_config.h:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Взаимодействие между рабочими организовано согласно паттерну producer-consumer:

Рабочик на 1 участке создаёт булавку, проверяет её на кривизну (спит в течение некоторого рандомного времени), и, если она не кривая, отправляет в циклическую очередь рабочим на втором участке.

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

Рабочий на втором участке ожидает и принимает не искревлённую булавку, после этого затачивает её (спит в течение некоторого времени), и после этого посылает её рабочим на 3 участке.

Изображение выглядит как текст, снимок экрана, дисплей

Автоматически созданное описание

Рабочий на 3 участке ожидает и принимает заточенную булавку, проверяет качество работы (спит в течение некоторого рандомного времени), и после этого пишет результат провеки в выходной поток stdout.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Логика работы с разделямой памятью реализована в функциях wait\_for\_pin и send\_pin:

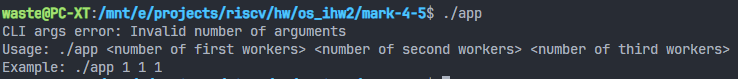
Изображение выглядит как текст, снимок экрана

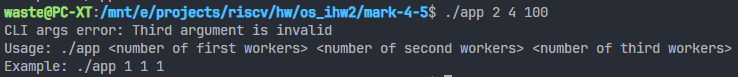
Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

Примеры работы программы:

Если входные аргументы командной строки указаны неверно, об этом сообщается вместе с подсказой и примером использования: 



Запуск программы с 3 рабочими на 1 участке, 5 рабочими на 2 участке и 2 рабочими на 3 участке:

Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание

Завершение работы программы при нажатии ctr+c: Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание