

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный ядерный исследовательский университет «МИФИ»
(НИЯУ МИФИ)
Институт Интеллектуальных Кибернетических Систем
Кафедра Кибернетики

Лабораторная работа №4

По курсу «Разработка программного обеспечения ОС UNIX»

Работу выполнил студент группы Б17-511:

Павлов И.Е.

Проверил:

Ктитров С.В.

Москва 2020

Постановка задачи

Тема «Семафоры POSIX»

Разработать программу, реализующую интерактивную работу с семафорами, предоставляющую возможность пользователю в диалоге изменять значение семафора, открывать, создавать, удалять, ожидать и т.п.

Объяснение логики

При запуске программы пользователю предлагается создать семафор, либо подключиться к существующему.

После подключения пользователь может управлять семафором, вводя соответствующий символ.

Вот так выглядит меню:

's' - показать значение
'd' - удалить семафор
'w' - уменьшить значение (ожидать с блокировкой)
't' - попробовать уменьшить
'f' - попробовать уменьшить (ждать определенное время)
'p' - увеличить значение
'e' - выход (семафор будет просто отсоединен от процесса, но НЕ удален!)

Пример работы

В ходе выполнения работы было обнаружено, что функция `sem_getvalue` не поддерживается системой Mac OS X 10.6.8. Поэтому примеры запущены на Debian.

В примере запущены два экземпляра программы и разыгран следующий сценарий:

Первая программа создает семафор “/mysem” со значением = 2. Вторая программа подключается к этому семафору, считывает значение, затем дважды уменьшает. После этого первая программа сначала пробует уменьшить (не получается), затем блокируется в ожидании возможности уменьшить. После вторая программа увеличивает значение (и сразу после первая программа уменьшает значение и разблокируется). После первая программа отсоединяется от семафора (без удаления). Вторая программа удаляет семафор.

```
Activities Terminal user@debian: /media/sf_sh Sat 15:30 user@debian: /media/sf_sh en
File Edit View Search Terminal Help
user@debian:/media/sf_sh$ ./sem
Вы хотите создать новый семафор? [y (новый) / n (уже существующий)] -> y
Введите название семафора (начните со знака '/') -> /mysem
Введите значение семафора -> 2
Создали
Введите:
's' - показать значение
'd' - удалить семафор
'w' - уменьшить значение (ожидать с блокировкой)
't' - попробовать уменьшить
'f' - попробовать уменьшить (ждать определенное время)
'p' - увеличить значение
'e' - выход (семафор будет просто отсоединен от процесса, но НЕ удален!)
-> t
Не получилось уменьшить
Введите:
's' - показать значение
'd' - удалить семафор
'w' - уменьшить значение (ожидать с блокировкой)
't' - попробовать уменьшить
'f' - попробовать уменьшить (ждать определенное время)
'p' - увеличить значение
'e' - выход (семафор будет просто отсоединен от процесса, но НЕ удален!)
-> w
Сейчас значение семафора = 0, процесс будет блокирован, пока не появится возможность уменьшить значение
Уменьшили
Введите:
's' - показать значение
'd' - удалить семафор
'w' - уменьшить значение (ожидать с блокировкой)
't' - попробовать уменьшить
'f' - попробовать уменьшить (ждать определенное время)
'p' - увеличить значение
'e' - выход (семафор будет просто отсоединен от процесса, но НЕ удален!)
-> e
Отсоединили семафор
Пока!
user@debian:/media/sf_sh$

File Edit View Search Terminal Help
user@debian:/media/sf_sh$ ./sem
Вы хотите создать новый семафор? [y (новый) / n (уже существующий)] -> n
Введите название семафора (начните со знака '/') -> /mysem
Подключились
Введите:
's' - показать значение
'd' - удалить семафор
'w' - уменьшить значение (ожидать с блокировкой)
't' - попробовать уменьшить
'f' - попробовать уменьшить (ждать определенное время)
'p' - увеличить значение
'e' - выход (семафор будет просто отсоединен от процесса, но НЕ удален!)
-> s
2
Уменьшили
Введите:
's' - показать значение
'd' - удалить семафор
'w' - уменьшить значение (ожидать с блокировкой)
't' - попробовать уменьшить
'f' - попробовать уменьшить (ждать определенное время)
'p' - увеличить значение
'e' - выход (семафор будет просто отсоединен от процесса, но НЕ удален!)
-> w
Уменьшили
Введите:
's' - показать значение
'd' - удалить семафор
'w' - уменьшить значение (ожидать с блокировкой)
't' - попробовать уменьшить
'f' - попробовать уменьшить (ждать определенное время)
'p' - увеличить значение
'e' - выход (семафор будет просто отсоединен от процесса, но НЕ удален!)
-> w
Уменьшили
Введите:
's' - показать значение
'd' - удалить семафор
'w' - уменьшить значение (ожидать с блокировкой)
't' - попробовать уменьшить
'f' - попробовать уменьшить (ждать определенное время)
'p' - увеличить значение
'e' - выход (семафор будет просто отсоединен от процесса, но НЕ удален!)
-> p
Увеличили
Введите:
's' - показать значение
'd' - удалить семафор
'w' - уменьшить значение (ожидать с блокировкой)
't' - попробовать уменьшить
'f' - попробовать уменьшить (ждать определенное время)
'p' - увеличить значение
'e' - выход (семафор будет просто отсоединен от процесса, но НЕ удален!)
-> d
Удалили
Пока!
user@debian:/media/sf_sh$
```

Листинг программы

```
#include <semaphore.h>      /* sem_open(), sem_destroy(), sem_wait().. */
#include <fcntl.h>          /* O_CREAT, O_EXCL */
#include <iostream>
#include <limits.h>

int main(int argc, char *argv[])
{
    char name[_POSIX_NAME_MAX];
    int value;
    char ch;
    sem_t *sem;

    std::cout << "Вы хотите создать новый семафор? [у (новый) / н (уже существующий)] -> ";
    std::cin >> ch;

    switch (ch)
    {
        case 'y':
            std::cout << "Введите название семафора (начните со знака '/') -> ";
            std::cin >> name;
            std::cout << "Задайте значение семафора -> ";
            std::cin >> value;
            sem = sem_open(name, O_CREAT | O_EXCL, 0666, value);
            if (sem != SEM_FAILED)
                std::cout << "Создали" << std::endl;
            else
            {
                std::cout << "Не удалось создать" << std::endl;
                return 0;
            }
            break;

        case 'n':
            std::cout << "Введите название семафора (начните со знака '/') -> ";
            std::cin >> name;
            sem = sem_open(name, 0);
            if (sem != SEM_FAILED)
                std::cout << "Подключились" << std::endl;
            else
            {
                std::cout << "Не удалось подключиться" << std::endl;
                return 0;
            }
            break;

        default:
            std::cout << "Введено неверное значение" << std::endl;
            return 0;
    }

    bool keepGoing = true;
    while (keepGoing)
    {
        std::cout << std::endl << "Введите:" << std::endl << "'s' - показать значение" <<
std::endl << "'d' - удалить семафор" << std::endl << "'w' - уменьшить значение (ожидать с
блокировкой)" << std::endl << "'t' - попробовать уменьшить" << std::endl << "'f' - попробовать
уменьшить (ждать определенное время)" << std::endl << "'p' - увеличить значение" << std::endl <<
"'e' - выход (семафор будет просто отсоединен от процесса, но НЕ удален!)" << std::endl << " ->
";

        std::cin >> ch;

        switch (ch)
        {
            case 's':
                if (!sem_getvalue(sem, &value))
                    std::cout << value << std::endl;
                break;
        }
    }
}
```

```

    case 'd':
        if(!sem_unlink(name))
        {
            std::cout << "Удалили" << std::endl;
            keepGoing = false;
        }
        break;

    case 'w':
        sem_getvalue(sem, &value);
        if (value == 0)
            std::cout << "Сейчас значение семафора = 0, процесс будет блокирован, пока
не появится возможность уменьшить значение" << std::endl;
        if (!sem_wait(sem))
            std::cout << "Уменьшили" << std::endl;
        break;

    case 't':
        if (!sem_trywait(sem))
            std::cout << "Уменьшили" << std::endl;
        else
            std::cout << "Не получилось уменьшить" << std::endl;
        break;

    case 'f':
        struct timespec time;
        std::cout << "Введите количество секунд ожидания: " << std::endl;
        std::cin >> time.tv_sec;
        if (!sem_timedwait(sem, &time))
            std::cout << "Уменьшили" << std::endl;
        else
            std::cout << "Время истекло, не получилось уменьшить" << std::endl;
        break;

    case 'p':
        if (!sem_post(sem))
            std::cout << "Увеличили" << std::endl;
        else
            std::cout << "Не получилось увеличить" << std::endl;
        break;

    case 'e':
        if (!sem_close(sem))
            std::cout << "Отсоединили семафор" << std::endl;
        keepGoing = false;
        break;

    default:
        std::cout << "Введено неверное значение" << std::endl;
}
}

std::cout << "Пока!" << std::endl;

return 0;
}

```