Задания (модуль 1)

Kirill Krinkin edited this page · 6 revisions

001 Динамическая линковка

Разработать программу "Hello , World!", печатающую приветствие на консоли. Функцию печати поместить в отдельную динамическую библиотеку (shared object, .so)

002 Чтение переменных окружения

Разработать программу, выводящую на экран значение теекущего каталога из которого запущена программа, с использованием чтения переменных окружения из массива, передаваемого в функцию main или с использованием функции getenv

003 Загрузка динамических библиотек в runtime с libdl

Для программы, зазработанной в 001, заменить линамическую линковку функции печати из разделяемой библиотеки на загрузку в runtime. Выполнить задание с использованием библиотеки libdl. (См. функции dlopen, dlsym).

004 Построение дерева процессов

Пользуясь информацией, размещенной в файловой системе /proc прочитать и вывести на консоль информацию о дереве процессов в системе.

(!) Важно: корректно обрабатывать ситуации, в которых отсутствуют права на чтение содержимого файлов и каталогов.

Программа должна принимать на входе 1 параметр с идентификатором процесса, для которого строится список родителей.

(*) Если параметр отсутствует, то необходимо вывести дерево для всех процессов в системе, к информации о которых у пользователя есть доступ.

005 Shell

Разаботать командный интерпретатор, обладающий следующими чертами:

- запуск процессов, набранных пользователем в командной строке
- в отличие от стандартного командного интерпретатора, поиск исполняемых файлов производится не только в переменной PATH, но и в текущем каталоге
- выход из интерпретатора должен осуществляться по Ctrl-C, перед выходом, командный интерпретатор должен напечатать на экране "Goodbye" (обработчик сигнала SIGTERM долженбыть перегружен).

006 mykill

Разработать программу отправляющую конкретный сигнал заданному процессу. Формат вызова:

```
./mykill 1231 15
```

где первый параметр -- идентификатор процесса, второй параметр -- десятичный номер сигнала.

007 summator

Разработать приложение с использованием pthreads для вычисления в многопоточном режиме интеграла функции y=sin(x) на интервале [0..N] Программа должна принимать 2 аргумента N и n обоззначающие соответственно:

- N - правая граница интервала
- n - количество потоков

008 thread list

Разработать программу выводящую список потоков процесса заданного в качестве аргумента.

009 pipe readers

Разработать программу, которая читает вывод консольной программы переданной ей в качестве аргумента, и выводит на консоль инвертированные строки.

010 tie-tie

Разработать программу, которая может встраиваться в цепочку вызовов proc1 | tie-tie **string filename** | proc2 |... , подсчитывать количество вхождений строки **string**, и выводить это значение в файл **filename**

011 pipe chat

Разработать программу, реализующую функцию чата для двух взаимодействующих процессов. Программа должна иметь следующий формат вызова:

```
./mychat pipein pipeout
```

Для передачи сообщений должен использоваться механизм именованных каналов. Если при запуске программы, указанные в качестве тринспорта каналы не существуют, то они должны быть созданы.

012 readers/writers

Реализовать многопоточную программу выполняющую конкурентную запись/чтение данных в контейнер (vector) в памяти с периодическим выводом количества элементов на консоль.

Формат вызова

```
./rwqueue writers readers max
```

Где:

- writers - количество писателей
- readers - количество читателей
- max - максимальное количество элементов в контейнере

Выход из программы должен осуществляться с корректным завершением всех потоков по нажатию Ctrl+C

013 udp echo

Реализовать функции чат-программы разработанной в **011** заменив транспортный уровень на протокол UDP, работающий на локальном адресе.

Формат вызова

```
./udpchat localport remoteport
```

014 broadcast

Разработать программу реализующую групповой чат в рамках локальной сети на основе броадкаст сообщений.

015 tcp chat

Разработать приложения чат-клиента и чат-сервера, обеспечивающих следующие функции:

Сервер:

- подключение и отключение клиентов
- отправка списка подключенных клиентов по запросу в клиентское приложение
- Хранение общего списка сообщений
- отображение событий подключений/отключений

Клиент:

- подключение к серверу
- отключение от сервера
- отправка сообщений в общий чат
- получение последних N сообщений с сервера

Формат вызова:

- myserver port name
- myclient ip-address port name

016 static http

Реализовать статический web сервер, отдающий в качестве index.html страницу с именем автора и его фотографией, и корректно обрабатывающий ситуацию HTTP 404

Задания (модуль 2)

Kirill Krinkin edited this page · 4 revisions

017 clone

Переписать программу summator (задание 007) с использованием clone для создания потоков.

018 daemon

Переработать программу web-сервер(задание 016) Таким образом, чтобы в дополнение к указанным ключам командной строки поддерживался необязательный параметр -daemon (рекомендуется использованием функций getopt), запускающий сервер как процесс-демон. Демон должен писать в syslog при старте и при завершении работы. Завершение работы должно осуществляться по сигналу 15 (SIGTERM)

019 select

Переработать программу pipe chat (011) с тем, чтобы для операций чтения и записи использовался мультиплектор select. Программа должна быть однопоточной.

020 poll

Переработать программу pipe chat (011) с тем, чтобы для операций чтения и записи использовался мультиплектор poll. Программа должна быть однопоточной.

021 mmap sort

Разработать программу для сортировки большого файла, не загружая его в память (используя mmap). Программа должна поддерживать вызов:

./mmsort file size

где

- file -- сортируемый файл
- size число из [1..256] -- размер элемента данных для сортировки.

022

Переписать программу readers/writers (задание 012) таким образом, чтобы:

- буфер находился в разделяемой памяти
- Монитор выводящий состояние буфера выполнен в качестве отдельного приложения, которое печатает состояние буфера и завершает работу.

023 Atomic

отложен

024 Futex

отложен

025 ping

Реализовать утилиту ping с использованием raw сокетов.

026 tracert

Реализовать утилиту tracert с использованием raw сокетов.

027 ns_uts

Реализовать программу создающую процесс с собственным пространством имен UTS. В данном процессе поменять имя хоста с помощью sethostname и продемонстрировать разный результат выдаваемый uname. В качестве образца использовать шаблон demo_uts_namespaces.c. Создание потомка выполнить с использованием unshare.

028 ns_pid

Реализовать программу создающую иерархию процессов с собственными пространствами имен PID. В качестве образца использовать шаблон multi_pidns.c. Создание потомка выполнить с использованием unshare.

Задания (модуль 3)

Kirill Krinkin edited this page · 2 revisions

029 hellok

Разработать 2 модуля ядра. Первый модуль должен содержать функцию и экспортировать соответствующий символ для печати приветствия в syslog. Второй модуль должен вызывать функцию первого модуля при старте.

Модули должны быть оформлены в соответствии с https://www.kernel.org/doc/Documentation/kbuild/modules.txt как External Module

CP1: Контрольная точка 1

Kirill Krinkin edited this page · 2 revisions

Разработать программу демон, обрабатывающую сигналы пользователя SIG_USR1, SIG_USR2. Обработка сигналов заключается в подсчете их числа и сохранении в памяти процесса.

Демон использует сокет (протокол UDP или TCP) для того, чтобы:

- включать/выключать подсчет конкретного сигнала
- получать статистику по сигналам.

Исходный код приложения поместить в папку ср1 репозитария.