

Лабораторные работы по СПОВМ

Оглавление

1. Общие замечания.....	1
2. Лабораторные.....	2
2.1. Знакомство с Linux (Unix) и средой программирования.....	2
2.2. POSIX-совместимая файловая система.....	2
2.3. Лабораторная работа No 3. Понятие процессов.....	2
2.4. Лабораторная работа No 4. Взаимодействие и синхронизация процессов.....	3
2.5. Лабораторная работа No 5. Потоки исполнение (threads).....	3
2.6. Лабораторная работа No 6. Взаимодействие и синхронизация потоков.....	3
2.7. Лабораторная работа No 7. Задача производителя-потребители для процессов.....	3
2.8. Лабораторная работа No 8. Задача производителя-потребители для потоков.....	3
2.9. Лабораторная работа No 9. Работа с файлами, отображенными в память.....	3
2.10. Лабораторная работа No 10. Блокировки чтения/записи и условные переменные.....	3

1. Общие замечания

Требования к проекту лабораторной работы

Проект должен компилироваться и собираться gcc без предупреждений.

Опции gcc:

```
-std=c11 -pedantic -W -Wall -Wextra
```

Проект лабораторной работы должен располагаться в отдельном каталоге и содержать:

- исходные файлы на языке C с комментариями на русском языке (utf8);
- makefile – файл управления сборкой проекта;
- краткое описание проекта в текстовом формате (utf8);
- скрипты, входные или иные данные, необходимые для выполнения программ проекта;
- отчет о лабораторной работе (pdf).

Каталог проекта архивируется снаружи в .tar.gz, при разворачивании которого в текущем каталоге должна создаваться исходная файловая структура.

Наличие иных файлов, в том числе размещаемых операционной системой для целей индексации и прочих, не допускается. Примерный состав каталога с проектом lab03 и его архивом

```
$ find
.
./lab03.tar.gz
./lab03
./lab03/lab03.c
./lab03/consumer.c
./lab03/lab03.txt
./lab03/producer.c
./lab03/lab03.pdf
./lab03/makefile
$ tar tf lab03.tar.gz
lab03/
lab03/lab03.c
lab03/consumer.c
lab03/lab03.txt
lab03/producer.c
lab03/lab03.pdf
lab03/makefile
```

2. Лабораторные

2.1. Знакомство с Linux (Unix) и средой программирования

Оболочка `bash`, файловый менеджер `mc`, стандартное информационное обеспечение (`info`, `man`).

Внешнее знакомство с POSIX-совместимой файловой системой – структура каталогов, жесткие и символические ссылки, права доступа, монтирование файловых систем, монтирование каталогов (`mount`, `mount --bind`).

Команды и утилиты оболочки `man`, `info`, `mkdir`, `touch`, `rm`, `rmdir`, `cd`, `cat`, `sort`, `head`, `tail`, `tee`, `wc`, `chmod`, `pwd`, `ls`, `lsuf`, `lsblk`, `lsusb`, `lscpu`, `ln`, `link`, `unlink`, `locale`, `iconv`, `kill`, `top`, `htop`, `ps`, `grep`, `diff`, `env`, `file`, `stat`, `find`, `tar`, `gzip`, `more`, `less`, `printf`, `time`, ...

Сцепление программ и соединение выходных и входных стандартных потоков.

Перенаправление вывода `stdout` и `stderr` в файлы

Экосистема курса – `gcc`, `make`, `gdb`,

Задание

Освоить эффективную работу с файлами в оболочке и `mc`.

2.2. POSIX-совместимая файловая система

Структура ФС, содержимое `inode`, команды оболочки

Знакомство с POSIX-совместимой файловой системой – `opendir(3)`, `readdir(3)`, `closedir(3)`, `fstat(2)`, `readlink(2)`, `symlink(2)`, `link(2)`, `unlink(2)`, ...

Задание

Разработать программу `dirwalk`, сканирующую файловую систему и выводящую в `stdout` информацию в соответствии с опциями программы. Формат вывода аналогичен формату вывода утилиты `find`.

```
dirwalk [dir] [options]
```

`dir` – начальный каталог. Если опущен, текущий (`./`).

`options` – опции.

`-l` -- только символические ссылки (`-type l`)

`-d` -- только каталоги (`-type d`)

`-f` -- только файлы (`-type f`)

`-s` – сортировать выход в соответствии с `LC_COLLATE`

Если опции `ldf` опущены, выводятся каталоги, файлы, ссылки, как у `find` без параметров.

2.3. Лабораторная работа No 3. Понятие процессов.

Понятие процессов. `fork`, `execve`, `getenv`, `setenv`.

- 2.4. Лабораторная работа No 4. Взаимодействие и синхронизация процессов
 - 2.5. Лабораторная работа No 5. Потоки исполнение (threads)
 - 2.6. Лабораторная работа No 6. Взаимодействие и синхронизация потоков
 - 2.7. Лабораторная работа No 7. Задача производителя-потребители для процессов
 - 2.8. Лабораторная работа No 8. Задача производителя-потребители для потоков
 - 2.9. Лабораторная работа No 9. Работа с файлами, отображенными в память
 - 2.10. Лабораторная работа No 10. Блокировки чтения/записи и условные перемен-
- ные