

Задания к лабораторной работе №1 по курсу “Операционные системы”

1. Утилита `ps` при вызове выводит список задач текущего пользователя. Она печатает идентификационный номер задачи (PID), имя вызвавшей выполнение задачи программы и некоторую другую информацию. Утилита `kill` позволяет уничтожить задачу по ее идентификационному номеру, введенному в качестве параметра. Используя названные утилиты, написать программу `zap`, которая уничтожает задачу по имени, введенному в качестве параметра, породившей ее программы. Если заданному имени будет соответствовать несколько задач, то уничтожить первую из них.
2. Написать функцию для вычисления чисел Фибоначчи.
3. Написать программу-фильтр `overwrite`, которая направляет входной поток во временный файл. Когда поток входных данных иссякнет, временный файл переименовывается в имя, заданное параметром `overwrite`. Например, вызов

```
sort data.txt | overwrite data.txt
```

должен означать следующую последовательность операций: отсортированные строки файла `data.txt` помещаются во временный файл, временный файл замещает исходный файл `data.txt`. При вызове `overwrite` с двумя параметрами первый из них должен рассматриваться как имя программы, а второй как имя файла данных. Таким образом, должно быть возможным использовать вызов

```
overwrite sort data.txt
```

вместо приведенной выше “трубы”.

4. Написать программу `where`, которая рассматривает свой единственный параметр как имя программы и печатает каталоги, в которых эта программа расположена, или сообщает, что такой программы в системе нет. Например, вызов

```
where ls
```

должен обусловить печать каталога `/bin`. Нужно проверять все каталоги, определяемые переменной среды `PATH`.

5. Создать сценарий, решающий задачу вставки столбца в текстовую таблицу. Например, команда `ls -l` печатает подобную таблицу и нужно вставить в нее заданный в файле столбец в заданную позицию.

Задания к лабораторной работе №2 по курсу “Операционные системы”

1. Написать сценарий для `sed` для удаления тегов и комментариев HTML из заданного файла. Теги и комментарии могут быть многострочными. Обеспечить правильную подстановку для знаков `<`, `>`, `&` и `"`.
2. Как задание 1, но на `awk`.
3. Реализовать на Tcl/Tk диалоговое окно, обеспечивающее ввод арифметического выражения. Окно должно содержать кнопку “Счет”, по нажатию на которую в одном из элементов окна (не в строке ввода) должен появляться результат расчета. В окне также должна присутствовать кнопка “Выход”. Нужно обеспечить правильную привязку клавиш `Enter` и `Escape`. В окне должен быть элемент, меняющий цвет символов, когда курсор мышки находится над этим элементом.
4. Реализовать на `си++` классы “потребитель” и “производитель”, реализующие соответствующие процессы. Производитель должен генерировать случайные числа в максимально допустимом функцией `rand` диапазоне. Потребитель должен проверять числа на простоту, вести статистическую обработку (всего чисел обработано и доля простых) и печатать ее текущие результаты. Производитель должен после генерации 10000 чисел делать паузу на 1 секунду. Программа должна прекращаться по перехватываемому сигналу `SIGINT`. При реализации использовать функции `fork`, `read`, `write` и `pipe`. Сравнить результат с теоретическим, т.е. с теоретической долей простых чисел в заданном диапазоне.
5. Сделать на HTML форму, обеспечивающую ввод произвольного количества чисел через пробел (в элементе `TEXTAREA`). Форма должна содержать кнопку “Суммировать”, по нажатию на которую в одном из элементов формы (не `TEXTAREA`) должна появляться сумма введенных чисел. Подсчет суммы должен производиться CGI-сценарием, написанным с использованием `awk` и `bash`.