

Задание по курсу «Операционные системы»

осень 2016

1.
 - (a) Написать скрипт, который создает много (порядка 100 штук) файлов в вашей рабочей директории с различными именами, среди которых есть имена начинающиеся на «a», на «b» и на «c».
 - (b) Написать скрипт, который удаляет все файлы, имеющие имена начинающиеся на «a». (команда `rm`)
 - (c) Написать скрипт, который переименовывает все файлы с именами, начинающиеся на «b». (команда `mv`)
 - (d) Написать скрипт, который копирует под новым именем все файлы, имеющие имена начинающиеся на «c». (команда `cp`)
 - (e) Написать скрипт, который ищет последовательность введенных с клавиатуры символов в текстовом файле и выводит все строки, содержащие эту последовательность в файл. Текстовый файл для поиска выбрать или создать самостоятельно. (команда `grep`)
2. Реализовать (на Си) программу копирования файлов (назвав ее, например, `CP`), используя функции (системные вызовы) `read`, `write`, `open`, `close`, `creat`. Имена файлов должны задаваться как параметры командной строки. Желающие могут реализовать все или некоторые опции команды `cp` (далее текст `man`)
`cp` копирует файлы (или, если попросить, каталоги). Вы можете либо копировать один файл в другой, заданный файл, либо копировать сколько угодно файлов в заданный каталог.
Если последний аргумент является существующим каталогом, то `cp` копирует каждый исходный файл в этот каталог (сохраняя имена). В противном случае, если задано только два файла, то `cp` копирует первый файл во второй. Если задано более двух аргументов,

которые не являются опциями, а последний аргумент не является именем какого-либо каталога, то это приведет к ошибке.

(Так, если /a – это каталог, то `cp -r /a /b` скопирует /a в /b/a и /a/x в /b/a/x в случае, если /b уже существует, но эта же команда будет копировать /a в /b и /a/x to /b/x, если /b не существует, или же завершится с ошибкой, если /b – обычный файл).

Права доступа к файлам и каталогам будут равны тем, что были на оригинальных файлах, логически умноженными (AND) на 0777 (другими словами, в новых правах будут сброшены биты sticky, setuid и setgid – Прим. ред.), а также с учетом `umask(1)` пользователя (за исключением случая, когда задана опция `-p`). (Но во время рекурсивного копирования каталогов, вновь создаваемые каталоги будут временно получать права, логически сложенные (OR) со значением `S_IRWXU (0700)`, для того, чтобы разрешить чтение, запись и поиск во вновь созданных каталогах).

При попытке скопировать файл сам в себя ничего происходить не будет (за исключением возможной выдачи сообщения об ошибке). Когда происходит копирование файла в другой существующий файл, то он открывается с использованием вызова `open(path, O_WRONLY | O_TRUNC)`. Когда копирование осуществляется во вновь создаваемый файл, то он создается с использованием вызова `open(path, O_WRONLY | O_CREAT, mode)`. Если эти вызовы завершаются неудачно, то считается, что файл существует, а чтобы `cp` попытался его удалить (`unlink`), необходимо указать опцию `-f`. Если удаление проходит успешно, то дальше все работает как для случая с новым файлом.

ОПЦИИ POSIX

POSIX имеет четыре с половиной опции:

- f Если требуется, удаляет существующие файлы, в которые происходит копирование. (См. выше)
- i Спрашивает, нужно ли перезаписывать существующие файлы, в которые происходит копирование. (Запрос выдается на стандартный вывод ошибок, а ответ читается со стандартного ввода. Копирование осуществляется только в случае положительного ответа).
- p Сохраняет исходные параметры файла, такие как владелец, группа, права доступа (включая `suid` и `sgid` биты), время последней модификации и время последнего доступа к файлу.

В случае, если установка владельца или группы приводит к ошибке, `suid` и `sgid` биты сбрасываются. (Заметим, что впоследствии исходный файл и его копия могут иметь разное время последнего доступа, так как операция копирования является доступом к исходному файлу).

- R Копировать каталоги рекурсивно; правильно обрабатывать ситуации, когда попадают объекты, не являющиеся обычными файлами или каталогами. (Так, копией FIFO или специального файла будет также являться FIFO или специальный файл).
- r Копировать каталоги рекурсивно; выполнять какие-то, неопределенные стандартом действия, когда попадают объекты, не являющиеся обычными файлами или каталогами. (Так, разрешается, и фактически одобряется, наличие опции -r как синонима для -R. Однако, глупое поведение, которое проявляет версия `cp` в GNU 4.0 не запрещается.)

Замечание:

- (a) Подробное описание системных вызовов для работы с файлами см. гл. 8 в книге Кернигана и Ритчи «Язык программирования Си».
 - (b) При сборке программ на Си пользоваться утилитой `make`.
3. Требуется выполнить запуск процессов согласно графу (граф придумать самостоятельно).
Подсчитать количество процессов в графе и вывести в одном из процессов полученное значение. Для передачи информации между процессами использовать неименованные или именованные каналы (`pipe` и `FIFO`).
 4. Реализовать программу с тремя процессами, выполняющую сложение двух матриц. Один из процессов выполняет ввод исходных матриц в разделяемую память (например генерирует значения или выполняет ввод с терминала или из файла), другой выполняет сложение этих матриц третий выводит результат на терминал или в файл. Действия в процессах должны выполняться в бесконечном цикле до поступления команды завершения работы программы.
 5. Требуется выполнить запуск потоков (нитей исполнения) согласно графу в индивидуальном задании, посчитать количество запущенных по графу потоков и вывести полученное значение в последнем потоке.

6. Реализовать программу с тремя потоками, выполняющую сложение двух матриц. Один из процессов выполняет ввод исходных матриц (например генерирует значения или выполняет ввод с терминала или из файла), другой выполняет сложение этих матриц третий выводит результат на терминал или в файл. Действия должны выполняться в бесконечном цикле до поступления команды завершения работы программы.
7. Реализовать взаимодействующие по средствам сокетов процессы (клиент и сервер)
 - (a) использовать сокет с установлением логического соединения (TCP)
 - (b) сокет без установления логического соединения (UDP)

Клиенту при запуске в качестве аргумента командной строки передается адрес сервера. Клиент отправляет серверу сообщения, вводимые пользователем с консоли, сервер принимает эти сообщения, некоторым образом их обрабатывает (например изменяет регистр символов или переставляет символы в обратном порядке и т.п.) и отправляет обратно клиенту. Завершение клиентского процесса может происходить по вводу с консоли некоторого, заранее оговоренного, символа или с применением механизма сигналов.