

Отчёт о выполнении лабораторной работы №10

Российский Университет Дружбы Народов
Факультет Физико-Математических и Естественных Наук

Дисциплина: Операционные системы

Работу выполняла: Арежина Адриана

№ ст. билета: 1032201674

Группа: НКНбд-01-20

Москва. 2021г.









Цель работы

Изучить основы программирования в оболочке ОС UNIX. Научится писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

Задание

- Используя команды `getopts grep`, написать командный файл, который анализирует командную строку с ключами:
`--inputfile` — прочитать данные из указанного файла; `--outputfile` — вывести данные в указанный файл; `-r`-шаблон — указать шаблон для поиска; `--C` — различать большие и малые буквы; `-n` — выдавать номера строк.
А затем ищет в указанном файле нужные строки, определяемые ключом `-p`.
- Написать на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю. Затем программа завершается с помощью функции `exit(n)`, передавая информацию в о коде завершения в оболочку. Командный файл должен вызывать эту программу и, проанализировав с помощью команды `$?`, выдать сообщение о том, какое число было введено.
- Написать командный файл, создающий указанное число файлов, пронумерованных последовательно от `1` до `N` (например `1.tmp`, `2.tmp`, `3.tmp`, `4.tmp` и т.д.). Число файлов, которые необходимо создать, передаётся в аргументы командной строки. Этот же командный файл должен уметь удалять все созданные им файлы (если они существуют).
- Написать командный файл, который с помощью команды `tar` запаковывает в архив все файлы в указанной директории. Модифицировать его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (использовать команду `find`).

Выполнение работы

- Используя команды `getopts grep`, написала командный файл, который анализирует командную строку с ключами:
`--inputfile` — прочитать данные из указанного файла; `--outputfile` — вывести данные в указанный файл; `-r`-шаблон — указать шаблон для поиска; `--C` — различать большие и малые буквы; `-n` — выдавать номера строк. (см. рисунки ниже [файл 1](#), [файл 1.2](#))


- Написала на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю. Затем программа завершается с помощью функции `exit(n)`, передавая информацию в о коде завершения в оболочку. (см. рисунок ниже [программа](#))

- Создала командный файл, который вызывает эту программу и, проанализировав с помощью команды `$?`, выдаёт сообщение о том, какое число было введено. (см. рисунок ниже [файл 2](#))

- Написала командный файл, создающий указанное число файлов, пронумерованных последовательно от `1` до `N` (например `1.tmp`, `2.tmp`, `3.tmp`, `4.tmp` и т.д.). Число файлов, которые необходимо создать, передаётся в аргументы командной строки. Этот же командный файл умеет удалять все созданные им файлы (если они существуют). (см. рисунки ниже [файл 3](#), [файл 3.1](#), [результат](#))



- Написала командный файл, который с помощью команды `tar` запаковывает в архив все файлы в указанной директории. Модифицировала его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (использовала команду `find`). (см. рисунок ниже [файл 4](#))


Контрольные вопросы

- Она осуществляет синтаксический анализ командной строки, выделяя флаги, и используется для объявления переменных.
- При генерации имен файлов используют метасимволы:

- * - произвольная (возможно пустая) последовательность символов; - ? - один произвольный символ; - [...] - любой из символов, указанных в скобках перечислением и/или с указанием диапазона; - cat f* - выдаст все файлы каталога, начинающиеся с "f"; - cat *f* - выдаст все файлы, содержащие "f"; - cat - выдаст файлы данного каталога с однобуквенными расширениями, - program.? - скажем "program.c" и "program.o", но не выдаст "program.com"; - cat [a-d]* - выдаст файлы, которые начинаются с "a", "b", "c", "d". Аналогичный эффект дадут и команды "cat [abcd]*" и "cat [bdac]*".

3. for, case, if, while.
4. Break, continue.
5. Команды ОС UNIX возвращают код завершения, значение которого может быть использовано для принятия решения о дальнейших действиях. Команда test, например, создана специально для использования в командных файлах. Единственная функция этой команды заключается в выработке кода завершения.
6. Означает условие существования файла man\$s/\$i.\$s.
7. Если речь идет о 2-х параллельных действиях, то это while. когда мы показываем, что сначала делается 1-е действие. потом оно заканчивается при наступлении 2-го действия, применяем until.

Вывод

Я изучила основы программирования в оболочке ОС UNIX и научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.