

СПО. Лабораторная работа №1

Shell скрипты

Задание

Напишите сценарий (shell-скрипт), решающий задачу по номеру своего варианта. Скрипт должен иметь право на исполнение и запускаться без указания имени оболочки-интерпретатора.

Варианты заданий

1. Сценарий, который будет читать текст из заданного файла, и выводить, построчно, на `stdout`, добавляя при этом дополнительные пустые строки так, чтобы в результате получился вывод с двойным межстрочным интервалом. Добавьте код, который будет выполнять проверку наличия файла, передаваемого как аргумент. Когда сценарий будет отлажен, измените его так, чтобы он выводил текстовый файл с тройным межстрочным интервалом. И наконец, напишите сценарий, который будет удалять пустые строки из заданного файла.
2. Сценарий, который будет выводить себя на `stdout`, но в обратном порядке.
3. Для каждого файла из заданного списка, сценарий должен определить тип архиватора, которым был создан тот или иной файл (с помощью утилиты `file`). Затем сценарий должен выполнить соответствующую команду разархивации (`gunzip`, `bunzip2`, `unzip`, `uncompress` или что-то иное). Если файл не является архивом, то сценарий должен оповестить пользователя об этом и ничего не делать с этим файлом.
4. Сценарий должен сгенерировать “уникальный” 6-ти разрядный шестнадцатиричный идентификатор системы. Не пользуйтесь дефектной утилитой `hostid`. Подсказка: `md5sum /etc/passwd`, затем отберите первые 6 цифр.

5. Сценарий должен создать архив `*.tar.gz` всех файлов в домашнем каталоге пользователя, которые изменялись в течение последних 24 часов. Подсказка: воспользуйтесь утилитой `find`.
6. Сценарий должен вывести на `stdout` все простые числа, в диапазоне от 60000 до 63000. Вывод должен быть отформатирован по столбцам (подсказка: воспользуйтесь командой `printf`).
7. Сценарий должен имитировать работу лототрона – извлекать 5 случайных неповторяющихся чисел в диапазоне 1 - 50. Сценарий должен предусматривать как вывод на `stdout`, так и запись чисел в файл, кроме того, вместе с числами должны выводиться дата и время генерации данного набора.
8. Замена существующего расширения в имени файла на заданное. Исходное имя файла и новое расширение передаются скрипту в качестве параметров. Основное средство: нестандартное раскрытие переменных. Усложнение: предусмотреть штатную реакцию на отсутствие расширения в исходном имени файла.
9. Выдача сразу трех временных штампов для заданного имени файла. Основные средства: переменные оболочки, `gawk`, командная подстановка. Усложнение: обработка произвольного числа имен, реакция на ошибки.
10. Выделение из исходной строки подстроки с границами, заданными порядковыми номерами символов в исходной строке. Усложнение: предусмотреть возможность не выделения, а удаления подстроки. Основные средства: команда `cut`, переменные оболочки.
11. Добавление в конец каждого текстового файла текущего временного штампа. Имена текстовых файлов передаются скрипту в качестве параметров, их число заранее не известно. Основные средства: `for`, переадресация.
12. Организация таймера — периодическая выдача на `stdout` сообщения о том, сколько времени прошло после запуска таймера (т.е. скрипта) и сколько осталось до конца работы. Параметры таймера передать при запуске скрипта через параметры. Основные средства: `while` или `until`, средства для организации счетчика.

← ПРЕДЫДУЩАЯ

СЛЕДУЮЩАЯ →

СПО. Допуск к лабораторным работам

СПО. Лабораторная работа №2
