Список задач к семинарам по ОС.

Предисловие

Для выполнения заданий разрешается установка любых кастомных пакетов в систему любым менеджером пакетов: apt, DNF, YUM, RPM, snap, synaptic, etc. НО крайне рекомендую использовать CLI менеджеры, не GUI (как например, aptitude). Я собрал задачи по группам, но не по сложности. В процессе выполнения всех задач одной категории вы можете обнаружить, что уже умеете выполнять часть задач из другой категории. (Например, для выполнения задач по выводу конкретных параметров CPU вам может понадобится grep) Тем не менее, рекомендую делать такие задания повторно, чтобы лучше запомнить, как работать с той или иной командой.

В заданиях 100% будет то, что я вам не рассказывал, не пугайтесь, это нормально. Задачи в том числе нацелены на то, чтобы вы учились искать решения на любого рода задачи самостоятельно. На следующих парах мы будем разбирать сами задания. Будет еще одна возможность задать мне вопросы: как и что сделать.

Если какое-то задание вам не понятно, вызывает непреодолимые трудности или в задании есть не точность – пишите мне, будем разбираться.

Формат выполнения

Большинство задач предполагают написание команды в качестве ответа. Для некоторых же результатом может являться скрипт. Если в задании стоит вопрос, предполагающий на него текстовый ответ – пишем текст.

Таким образом ваша отчетность по этим задачам – список команд и краткий текстовый ответ на поставленные вопросы + архив со скриптами если вы их делали.

Язык программирования

Bash скриптинг имеет свой синтаксис, кроме того, поддерживает C-like код. Python – скриптовый язык по умолчанию. Кроме того, никто не мешает вам собрать из C++ кода приложение.

Все вышеперечисленное доступно для использования.

FYI:Ссылка на доки CLI в убунту

I wish you have fun!

Оглавление

System info

CPU

1. Какими способами можно узнать информацию о CPU?
    a. Написать минимум 2 способа
    b. Написать команды (или команду), которые выводят информацию о CPU. Список параметров для вывода:
        i. Название CPU
        ii. Число ядер/Потоков
        iii. Частота работы
        iv. Размер кэш-памяти (L1/2/3)
        v. Архитектуру CPU

RAM

2. Что такое RAM и Swap?
3. Написать команду, которая выводит объем RAM/Swap (любым способом)
4. Написать bash-script, который меняет размер Swap.
5. Для тех, у кого не гостевая Linux-система (т.е. основная). Написать команду, которая выводит информацию о типе используемой RAM (название, производитель, серийный номер, формат, объем, текущую частоту работы)

Disk Usage

6. Написать команду, которая выводит размер свободного месте на диске
7. Написать команду, которая выводит размер вашей home директории.
8. Написать команду, которая выводит список процессов

GPU (optional)

9. Вывести инфу о GPU (как правило команда зависит от типа GPU, необходима установка доп. пакетов. Хотя можно и системными средствами, хоть и не удобно)

Network

10. Написать команду, которая выводит IP-адресс компьютера
11. Написать команду, которая выводит MAC-адресс компьютера
12. Какие еще есть способы узнать эту информацию? Написать альтернативные варианты

Processes info

13. Вывести активные процессы в системе минимум двумя способами.

System Environment

14. Создать переменную окружения любым способом
15. Модифицировать PATH, добавив туда свою home директорию
16. Обновить имеющиеся системные библиотеки с помощью любого CLI менеджера пакетов
17. Для счастливых обладателей gpu: поставить на видеокарту драйвера, добавить пути к утилитам драйверов в окружение (обновить старые).
18. Создать alias к любой команде, сохранить его в конфигурационном файле
19. Поменять цвет терминала/текста в нем, размер окна, шрифт. На ваш вкус.

20. Для любителей питона: создать внутри своего home пространства virtual environment. Добавить в bashrc возможность его активировать.

Grep

Для работы с grep рекомендую выбрать достаточно крупный файл строк на 50 с осмысленной информацией, чтобы было что искать. Если не хочется думать над этим вопросом, загляните в задание #29

21. Что такое gpep?
    a. Общий синтаксис:
        i. Написать команду поиска паттерна в файле
        ii. Написать команду поиска паттерна в нескольких файлах сразу
        iii. Написать команду поиска нескольких паттернов в одном/нескольких файлах
22. Какой ключ делает команду не чувствительной к регистру?
23. Проверить с помощью grep наличие слова в файле. (т.е паттерн является словом целиком. Словоформы не в счет)
24. Написать команду вывода всех строк в файле, не содержащих паттерн.
25. Найти все info файлы в /proc директории с помощью grep
    a. Вывести их содержимое в терминал/файл
26. Добавить к команде из пункта 21.a.i номер строки, в которой нашлась подстрока
27. Поддерживает ли grep regex выражения?
    a. Прочитать tutorial
    b. Описать базовый regex синтаксис. (что означают ‘*’ ‘.’ ‘?’ ‘\’ и тд)

28. Написать команду вывода всех файлов с расширением ‘.so’ в директории /lib
    a. Без обхода поддиректорий
    b. С обходом поддиректорий
29. Скачать датасет.
    a. Отдельный + тому, кто разберется с kaggle API для загрузки датасетов и сделает это через терминал.
    b. Найти в Emails.csv все email с помощью grep
30. Модифицировать команду в п.29 так, чтобы считать паттерн из файла
31. Вывести процессы, запущенные системой, владельцем которых являетесь вы с помощью grep

Find

32. Написать команду поиска с заданным именем файла в заданной директории. Проверить работоспособность.
33. Написать команду поиска файлов по маске
34. Найти ключик, отвечающий за чувствительность к регистру
35. Найти все файлы с набором прав доступа 777 (или любой другой комбинацией)
36. Найти все файлы с доступом только на чтение
37. Найти все пустые директории
38. Найти все скрытые файлы
39. Найти все файлы, модифицированные за последние 3 часа/дня
40. Найти все файлы, которые были открыты за последние 3 часа / дня
41. Найти все файлы объемом от 10 до 15 Мбайт
42. Скопировать файлы из задания 32 (если они нашлись) в свою home директорию в поддиректорию tmp. (Find + cp)

43. Найти в home директории все скопированные файлы и удалить. (find + rm)

Bash

FYI: Cheat-Sheet

44. Создать директорию в своем home пространстве, где вы будете хранить все скрипты.
45. Написать «hello world» скрипт, запустить, убедиться, что он работает корректно.
46. Написать скрипт/функцию, который проверял бы существование файла или директории с заданным именем.
    a. Входные параметры:
        i. где искать
        ii. что искать (путь может быть относительным)
    b. Выход:
        i. Если файл или директория существуют:
            1. вывести тип найденного объекта (файл или директория)
            2. Абсолютный путь до искомого
        ii. Если файл или директория отсутствуют – вывести соответствующее сообщение
47. Написать функцию, которая проверяла бы ваши (*или любого другого пользователя) права доступа к заданному файлу или директории.
    a. Вход:
        i. Имя файла
        ii. Имя пользователя (если решили делать общий вариант)
    b. Выход:

i. Права доступа к файлу для вас (заданного пользователя)

48. На лабораторных по другим предметам, когда вы используете C/C++ вы часто пишите консольные приложения. Возьмите любую вашу лабу и сделайте из нее утилиту.
49. Написать скрипт, который бы:
    a. Определял ваш регион
    b. подключался к сервису погоды, брал погоду для вашего региона
    c. показывал ее в stdout / при указании файла в качестве параметра – сохранял бы результат в файл.
50. Обновлять информацию о погоде в файле, с помощью вашего скрипта, используя системный планировщик задач cron.
51. Написать скрипт, реализующий адресную книгу. В качестве хранилища информации можете использовать все что угодно: от текстового файла до БД. Скрипт должен поддерживать следующие функции
    a. Добавление контакта: придется определить поля для заполнения; например, имя фамилия, email, телефон, доп. информация.
    b. Удаление контакта
    c. Вывод информации о контакте по имени.
    d. Вывод списка контактов
    e. ** любые другие функции если вам интересно это задание. Это отличная возможность потренировать работу с БД.

Сделать так, чтобы вызов вашего скрипта/функции не содержал расширений файлов и вызова программной оболочки (как для системных команд). Например вместо

$$user@machine: \sim\$\ bash\ my\_simple\_command.sh$$

$$user@machine: \sim\$\ my\_simple\_command$$

52. Сделать так, чтобы функционал, описанный, в созданном вами файле, был доступен вам при входе в систему по умолчанию, без необходимости повторного добавления команд в программную оболочку вручную.
53. Модифицируйте все ваши скрипты, дав право на их исполнение всем пользователям. (но не на чтение и запись). Для себя оставить доступ на чтение и запись.
54. Создайте tar архив директории со скриптами, которые вы писали.

Administration

55. Создать нового пользователя. Без доп параметров
56. Добавить созданного пользователя в группу users. Удалить из группы администраторов или sudo если он там есть
57. Создать нового пользователя, без прав админа по умолчанию
58. Модифицировать процедуру создания пользователей, добавив в их окружение, написанные вами утилиты.
59. Модифицировать процедуру создания пользователей, так, чтобы при создании нового пользователя для него был создан virtual environment питона, в его home пространстве, а активация venv-а была доступна из окружения.