МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Тихоокеанский государственный университет»

Кафедра «Программное обеспечение вычислительной техники и автоматизированных систем»

Знакомство с операционной системой Linux

Лабораторная работа №1

по дисциплине «Операционные системы и системное ПО»

Выполнил студент Чекулаев В. Ю.

Факультет, группа ФКФН, ПО(аб)-81

Проверил Тормозов В. С.

Хабаровск – 2021г.

**Цель:** знакомство с ОС Linux.

**Задача:** получение первоначальных навыков работы в ОС Linux, изучение основных команд, изучение текстового редактора vi; изучение команд работы с файлами; изучение компилятора gcc.

1. **Описание выполнения работы и ответы на вопросы**

**Задание 1**

Ознакомился с командами useradd, passwd, exit, logout, who, shutdown, su, users, groups, а так же с командами, перечисленными в Таблице 1 лабораторной работы при помощи самой таблицы, мануалов (команда man), ключей -h или --help и сайта losst.ru.

*1) В чем отличие команды cat от less?*

Команда cat просто читает данные из файла или стандартного ввода и выводит их на экран. Тогда как команда less не считывает текст полностью, а загружает его небольшими фрагментами.

*2) Какие отличия имеются между командами exit и logout?*

Команда exit совершает выход из любого командного процессора, тогда как команда logout совершает выход только из командного процессора login. Если команду login вызвать не из командного интерпретатора, вызванного login, выводится сообщение и команда logout игнорируется.

*3) Как изменить права доступа к файлу?*

Для изменения прав доступа к файлу можно воспользоваться утилитой chmod следующим образом:

**$ chmod опции права /путь/к/файлу**

где опциями могут быть r — чтение, w — запись, x — выполнение, s - выполнение  от имени суперпользователя (дополнительный).

Также есть три категории пользователей, для которых можно установить эти права на файл: u - владелец файла, g - группа файла, o - все остальные пользователи.

Синтаксис настройки прав такой:

**группа\_пользователей действие вид\_прав**

В качестве действий могут использоваться знаки "+" - включить или "-" - отключить. Например:

**$chmod ugo+rwx file**

означает разрешить всем пользователям все права доступа к файлу file.

*4) Как создать учетную запись пользователя (в терминале)?*

Для создания учетной записи пользователя через терминал можно воспользоваться утилитой useradd следующим образом:

**$ useradd опции имя\_пользователя**

Основные опции:

* -b - базовый каталог для размещения домашнего каталога пользователя, по умолчанию /home;
* -c - комментарий к учетной записи;
* -d - домашний каталог, в котором будут размещаться файлы пользователя;
* -e - дата, когда учетная запись пользователя будет заблокирована, в формате ГГГГ-ММ-ДД;
* -f - заблокировать учетную запись сразу после создания;
* -g - основная группа пользователя;
* -G - список дополнительных групп;
* -k - каталог с шаблонами конфигурационных файлов;
* -l - не сохранять информацию о входах пользователя в lastlog и faillog;
* -m - создавать домашний каталог пользователя, если он не существует;
* -M - не создавать домашнюю папку;
* -N - не создавать группу с именем пользователя;
* -o - разрешить создание пользователя linux с неуникальным идентификатором UID;
* -p - задать пароль пользователя. В целях безопасности можно воспользоваться утилитой passwd уже после создания пользователя;
* -r - создать системного пользователя, не имеет оболочки входа, без домашней директории и с идентификатором до SYS\_UID\_MAX;
* -s - командная оболочка для пользователя;
* -u - идентификатор для пользователя;
* -D - отобразить параметры, которые используются по умолчанию для создания пользователя. Если вместе с этой опцией задать еще какой-либо параметр, то его значение по умолчанию будет переопределено.

**Задание 2**

Изучил описание редактора vi и основные его команды при помощи команды man, команды :help внутри самого редактора и при помощи сайта neoserver.ru.

*Какие режимы существуют в редакторе vi и чем они отличаются?*

**«Обычный режим»** — перемещение по файлу, стирание текста и другие редактирующие функции. Это - основной режим, только из него можно сразу перейти в другие режимы.

**«Режим ввода»** - ввод текста. Как только завершается ввод текста, принято сразу возвращаться в обычный режим. Заметьте, что стирание и ввод текста происходит в двух разных режимах. Переход в него из обычного режима:

i

<Insert>

***«Командный режим»*** -Команды (операции с файлом, поиск и замена, настройка редактора…). Переход в него из обычного режима осуществляется с помощью двоеточия (:).

**«Режим поиска»** - ввод поискового запроса. Переход в него из обычного режима

/ , поиск от курсора до конца документа

? , поиск от курсора до начала документа

***«Визуальный режим»*** - режим выделения текста:

"v" и влево или вправо стрелками;

Shift+v вся строка целиком;

Ctrl+v прямоугольник, часть текста.

**Задание 3**

*Как* *вывести на экран список файлов текущего каталога в краткой и расширенной форме?*

Для вывода информации в краткой форме достаточно вызвать утилиту ls без параметров. Для вывода информации в расширенной форме нужно добавить ключ -l к команде ls.

**Задание 4**

*Создайте каталог, удалите каталог. Создайте файл, отредактируйте его, скопируйте, а затем удалите файл. Опишите свои действия.*

Создать каталог можно при помощи команды mkdir, а удалить его при помощи команды rm -r.

Для того, чтобы создать файл можно воспользоваться редактором nano, а после открытия отредактировать его и сохранить изменения сочетанием клавиш Ctrl+O. Для копирования файла нужно воспользоваться командой cp, в которую первым аргументом передается имя исходного файла для копирования, а вторым имя выходного файла. Для удаления нужно воспользоваться командой rm.

*Как создать «жесткие» и «мягкие» ссылки на все файлы из какого-то определенного каталога?*

Для создания ссылок нужно воспользоваться утилитой ln следующим образом:

**$ ln опции файл\_источник файл\_ссылки**

Для создания «мягкой» (символьной) ссылки нужно воспользоваться опцией -s (softlink). А для создания жесткой ссылки нужно воспользоваться утилитой ln без параметров.

Вместо того, чтобы создавать ссылки на все фалы каталога можно создать ссылку на сам каталог. В таком случае будет легче обращаться к файлам, так как их исходные имена сохранятся.

**Задание 5**

*Определите, каким образом сконфигурирована корневая файловая система (команда mount) и сколько имеется свободного места (команда df) в каждой точке монтирования.*

Вывод команды mount на моем ПК:

1. sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
2. proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
3. udev on /dev type devtmpfs (rw,nosuid,relatime,size=3981296k,nr\_inodes=995324,mode=755)
4. devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
5. tmpfs on /run type tmpfs (rw,nosuid,noexec,relatime,size=809964k,mode=755)
6. /dev/sda5 on / type ext4 (rw,relatime,errors=remount-ro)
7. …

Из этого вывода, например, можно увидеть, что корневая файловая система располагается на разделе *SATA* диска */dev/sda5* и имеет формат файловой системы *ext4*, а в круглых скобках располагаются параметры монтирования (строка 6).

Вывод команды df на моем ПК:

Файл.система 1K-блоков Использовано Доступно Использовано% Cмонтировано в

udev 3981296 0 3981296 0% /dev

tmpfs 809964 2052 807912 1% /run

/dev/sda5 81570224 62068432 15315168 81% /

tmpfs 4049808 0 4049808 0% /dev/shm

tmpfs 5120 0 5120 0% /run/lock

tmpfs 4049808 0 4049808 0% /sys/fs/cgroup

…

Команда df дает практически те же данные, что и команда mount, только в более удобочитаемом виде. Плюс, здесь приводится информация о памяти файловых систем.

**Задание 6**

При создании программы я воспользовался редактором vi следующим образом:

**&vi main.c**

Далее, я перешел в режим ввода нажатием клавиши i и ввел следующий текст:

#include<stdio.h>

int main(){

printf(«Hello, world!\n»);

return 0;

}

После этого я перешел в командный режим нажатием клавиши Esc и ввел команду :x для записи изменений и выхода из редактора.

Далее, я откомпилировал эту программу и запустил ее с следующим образом:

**&gcc main.c -o main**

**&./main**

Вывод программы:

**Hello, world!**

*В чем отличие команд gcc и g++?*

Основные отличия:

1. gcc скомпилирует: \*.c\\*.cpp файлы как C и C ++ соответственно.
2. g++ скомпилирует: \*.c\\*.cpp файлы, но все они будут обрабатываться как файлы C ++.
3. Также, если вы используете g++ для связывания объектных файлов, он автоматически связывается в библиотеках std C ++ ( gcc не делает этого).
4. gcc Компиляция файлов C имеет меньше предопределенных макросов.
5. Gcc Компиляция \*.cpp и g++ компиляция \*.c\\*.cpp файлов имеет несколько дополнительных макросов.

Дополнительные макросы при компиляции \*.cpp файлов:

*#define \_\_GXX\_WEAK\_\_ 1*

*#define \_\_cplusplus 1*

*#define \_\_DEPRECATED 1*

*#define \_\_GNUG\_\_ 4*

*#define \_\_EXCEPTIONS 1*

*#define \_\_private\_extern\_\_ extern*

2. **Вывод**

В ходе данной лабораторной работы я получил первоначальные навыки работы в ОС Linux, изучил основные команды, изучил текстовый редактор vi; изучил команды работы с файлами; изучил компилятор gcc.