

## A. User Operations

1. List Users  
**less /etc/passwd**
2. Change User  
**su username**
3. Check Session  
**who**

## B. File Operations

1. Read File  
**cat filename**
2. Read File with Filtering  
**cat filename | grep "Text"**
3. Read File with Sorting  
**cat filename | sort**
4. Create a File  
**touch filename**  
**cat > filename**
5. Write Into A File  
**echo "Text" > filename**
6. Create Directory  
**mkdir name**
7. CHMOD  
**chmod -R 777 directory**
8. Remove File  
**rm -r filename**
9. Remove Directory  
**rmdir dir**
10. Edit File  
**vim filename**
11. Search File  
**Find -name "filename\*\*"**
12. Copy File  
**cp old new**
13. Move/Rename File  
**mv old new**

## C. Drives

1. See Data/File System Devices  
**df**
2. See Unmounted Disks  
**lsblk**

## D. Services

1. List Services Running  
**systemctl**
2. Services with Ports  
**ss -tulw**

## E. Compress / Extract

**tar -cvf newname files**

**tar -xvf name newdirectory**

## F. Ports

**ss -tulw**

**fuser 8080/tcp**

**fuser -k 8080/tcp**

## G. Folder Structure

**/etc => Configuration**

**/media**

**/usr**

**/lib**

**/opt**

**/home**

**ps -e | grep jenkins**

g++ main.cpp -o main;

```
#include <iostream>
using namespace std;
class Car {
protected:
    string Name;
    string Color;
public:
    virtual void see(){
        cout << "Name : " << this-
>Name << " Color: " << this-
>Color << endl;
    }
    void set(string a, string b){
        this->Name = a;
        this->Color = b;
    }
};
class FlyingCar :public Car{
private:
    double flyingSpeed = 0;
public:
    FlyingCar(double flyingSpeed){
        this->flyingSpeed = flyingSpeed;
    }
    void see() override{
        cout << "Name : " << this-
>Name << " Color: " << this->Color <<"
Speed: "<< flyingSpeed<< endl;
    }
};
int main(){
    Car car1;
    car1.set("Audi", "White");
    car1.see();
    FlyingCar car2(0.25);
    car2.set("Flying", "Sky Blue");
    car2.see();
    return 0;
}
```

## GTest Install / Config

sudo apt install libgtest-dev googletest

sudo cd /usr/src/gtest cmake -Bbuild

cmake --build build

Cd gtest\_main

Cp ./build/lib/libgtest\* /usr/lib

<https://www.youtube.com/watch?v=6dqFqh3UbPY>

## Main.cpp

```
#include <gtest/gtest.h>
int add(int a, int b){
    Return a+b;
}
TEST(add, positive){
    ASSERT_EQ(10, add(5,5));
}
Int main(int argc, char** argv){
    ::testing::InitGoogleTest(&argc, argv);
    RUN_ALL_TEST();
    Return 0;
}
```

## CMakeLists.txt

```
cmake_minimum_required(VERSION 3.22)
project("Gtest")
find_package(GTest REQUIRED)
add_executable(a.out main.cpp)
target_link_libraries(a.out
${GTEST_LIBRARIES})
enable_testing()
add_test(add a.out)
```

## Build

cmake -Bbuild

cmake --build build

cd build >> ctest -V