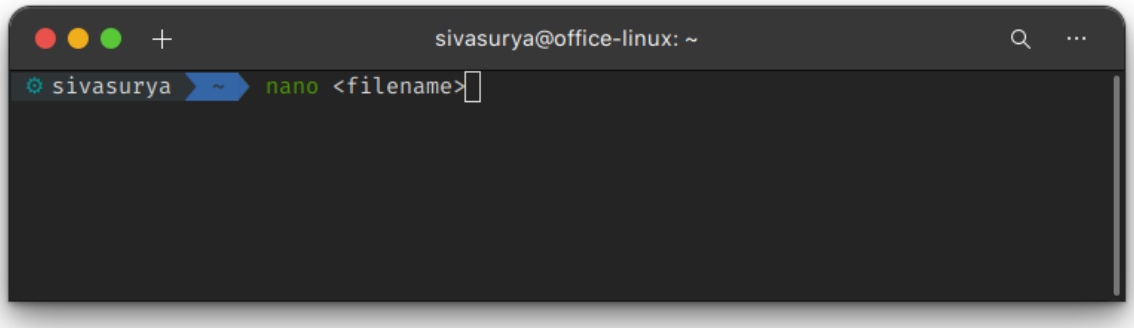
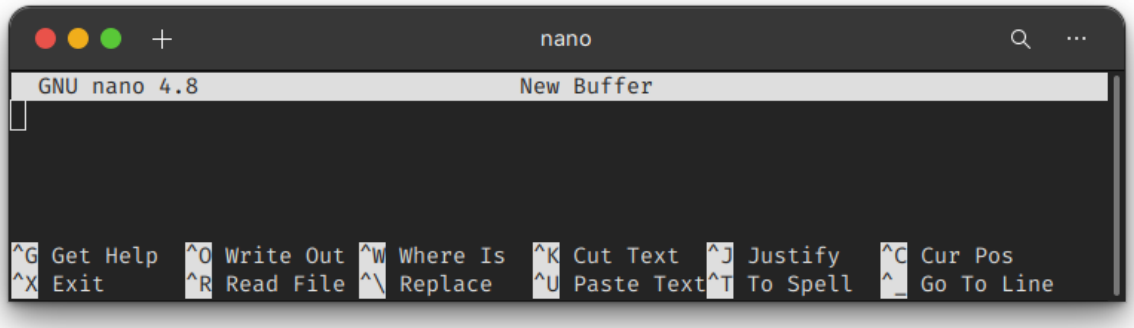
OS Lab Week 1

Siva Surya Babu
PES2201800475

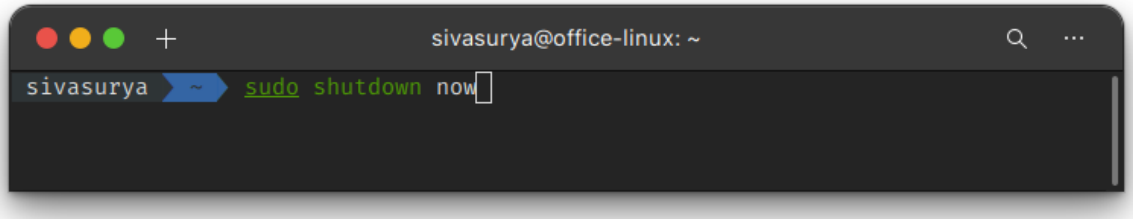
Task 1: Basic linux commands

1. Nano - simple command line text editor

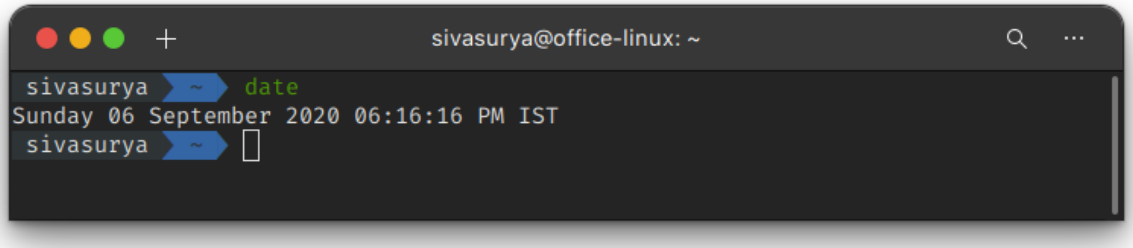




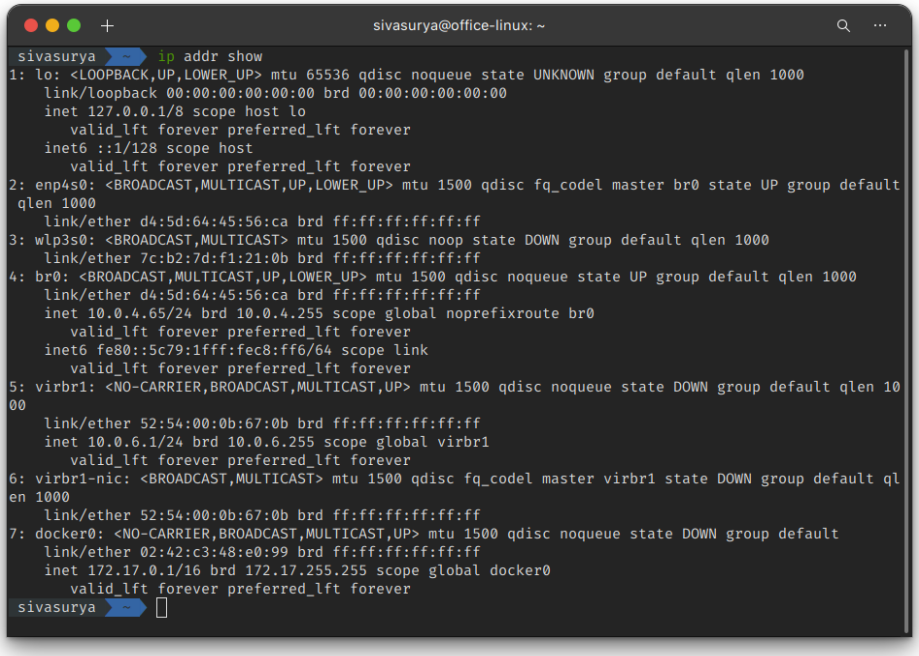
2. Shutdown - used to shutdown the system at a specified time with other options



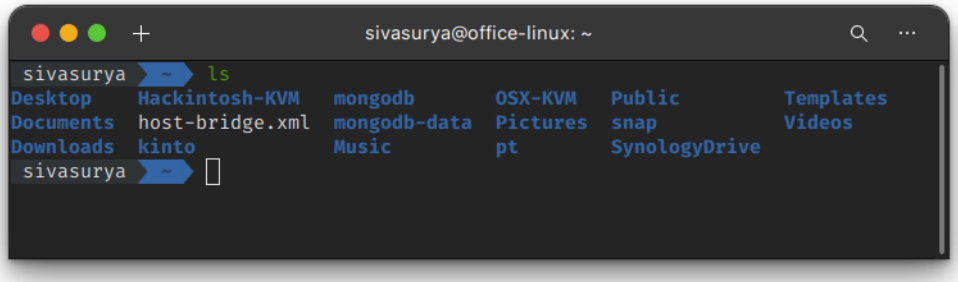
3. Date - displays the system date and time



4. Ip addr show - Display IP Addresses and property information



5. Ls - lists directories



Task 2: Write and compile a C program to display an array in reverse using index

Client.c

```c
#include <stdio.h>
#include "Header.h"

int main(int argc, char const *argv[])
{
    puts("enter the arr count");
    int count;
    scanf("%d", &count);
    puts("enter elements of arr");
    int arr[4];
    for (int i = 0; i < count; i++)
    {
        scanf("%d", &arr[i]);
    }
    puts("Input arr is");
    print_arr(count, arr);
    reverse_arr(count, arr);
    puts("Reversed arr is");
    print_arr(count, arr);
    return 0;
}
```

Header.c

```c
void reverse_arr(int count, int array[count]);
void print_arr(int count, int array[count]);
```

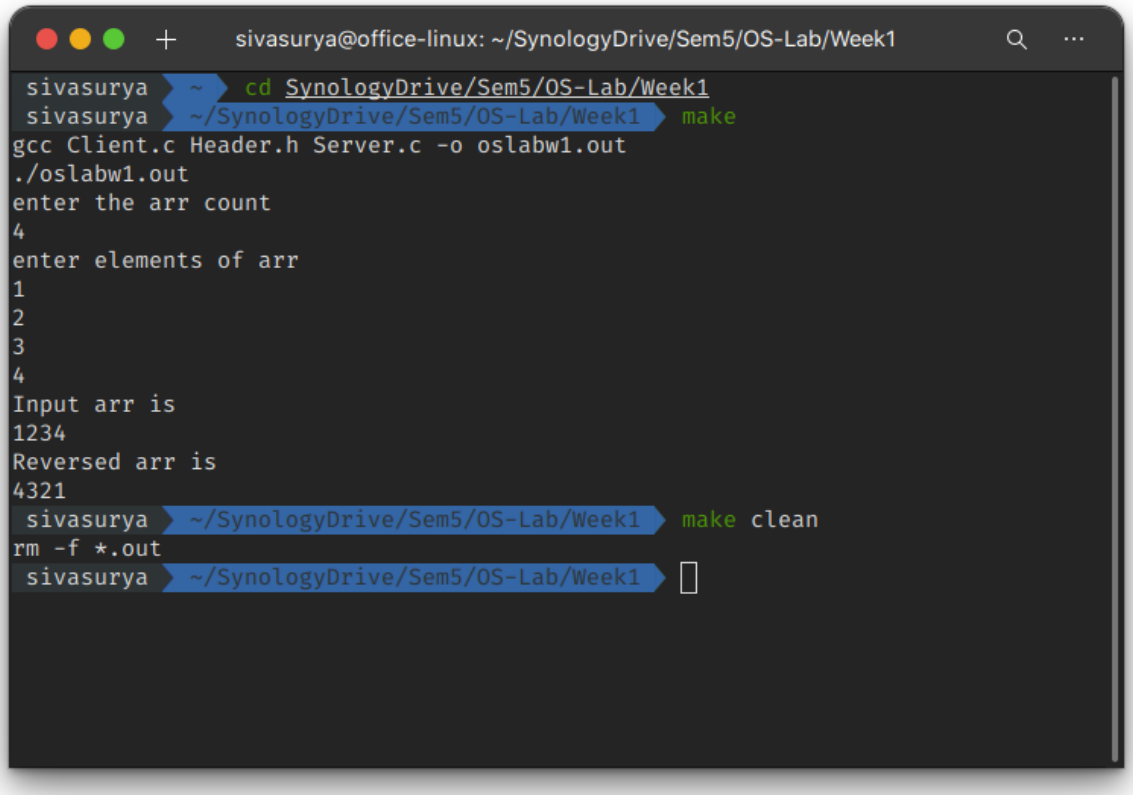
Server.c

```c
#include <stdio.h>

void print_arr(int count, int arr[count])
{
    for (int i = 0; i < count; i++)
        printf("%d", arr[i]);
    puts("");
}

void reverse_arr(int count, int arr[count])
{
    for (int i = 0; i < count / 2; i++)
    {
        int temp = arr[i];
        arr[i] = arr[count - i - 1];
        arr[count - i - 1] = temp;
    }
}
```

Makefile

```
oslabw1: Client.c Server.c
    gcc Client.c Header.h Server.c -o oslabw1.out
    ./oslabw1.out

clean:
    rm -f *.out
```

Output


sivasurya@office-linux: ~/SynologyDrive/Sem5/OS-Lab/Week1
sivasurya ~ cd SynologyDrive/Sem5/OS-Lab/Week1
sivasurya ~/SynologyDrive/Sem5/OS-Lab/Week1 make
gcc Client.c Header.h Server.c -o oslabw1.out
./oslabw1.out
enter the arr count
4
enter elements of arr
1
2
3
4
Input arr is
1234
Reversed arr is
4321
sivasurya ~/SynologyDrive/Sem5/OS-Lab/Week1 make clean
rm -f *.out
sivasurya ~/SynologyDrive/Sem5/OS-Lab/Week1

Task 3: Answer the following questions

1. **Why do we use makefile?**
   Make is a utility for managing compilation and execution of large programs with multiple files which require linking.
   The rules for make are placed in a file named Makefile which will contain the dependency rules and how to build files that are dependent on other files.
2. **Is makefile a shell script?**
   No, makefile is not a shell script.
3. **What does “clean” do in makefile**
   Clean is added to the makefile to remove the automatically generated object files and .out files
4. **How does make learn about the last modified files to be complied ?**
   It checks the time stamp of the file, if a dependency is newer than the target then the target gets rebuilt
5. **What does cflags in makefile mean ?**
   Cflag is an environment variable used to add arguments to the compiler
6. **Why do we use the -f option with make command ?**
   -f option is used to use a regular file as a makefile. Otherwise the make program looks for a file named makefile and use that