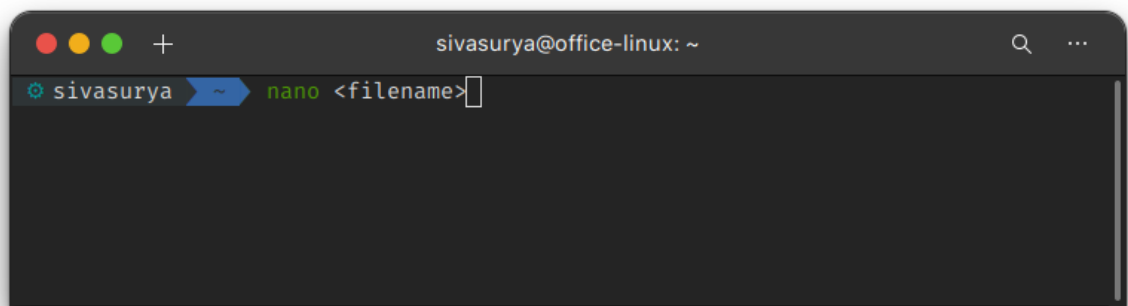


OS Lab Week 1

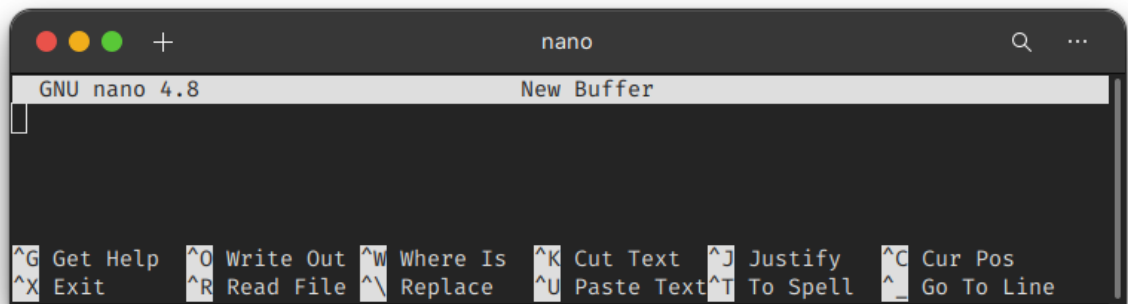
Siva Surya Babu
PES2201800475

Task 1: Basic linux commands

1. Nano - simple command line text editor

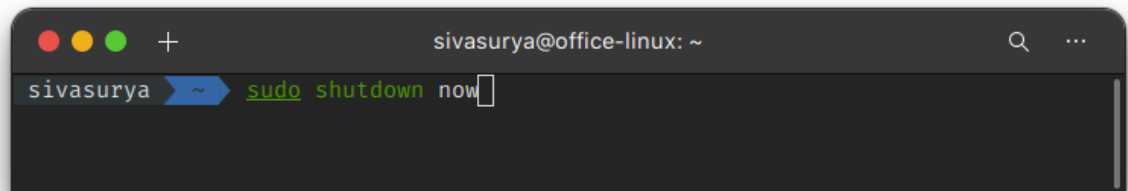


```
sivasurya@office-linux: ~  
sivasurya ~ nano <filename>
```



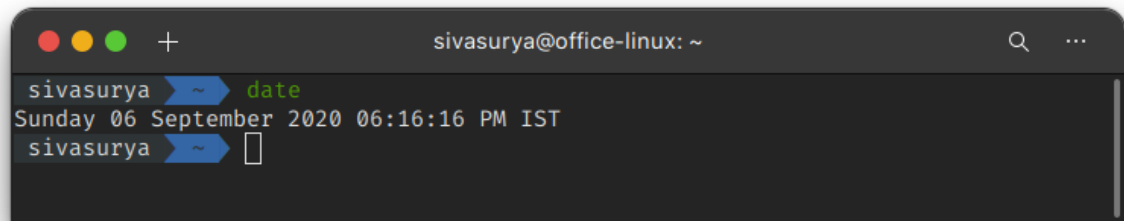
```
nano  
GNU nano 4.8 New Buffer  
  
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos  
^X Exit      ^R Read File ^\ Replace  ^U Paste Text ^T To Spell  ^_ Go To Line
```

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



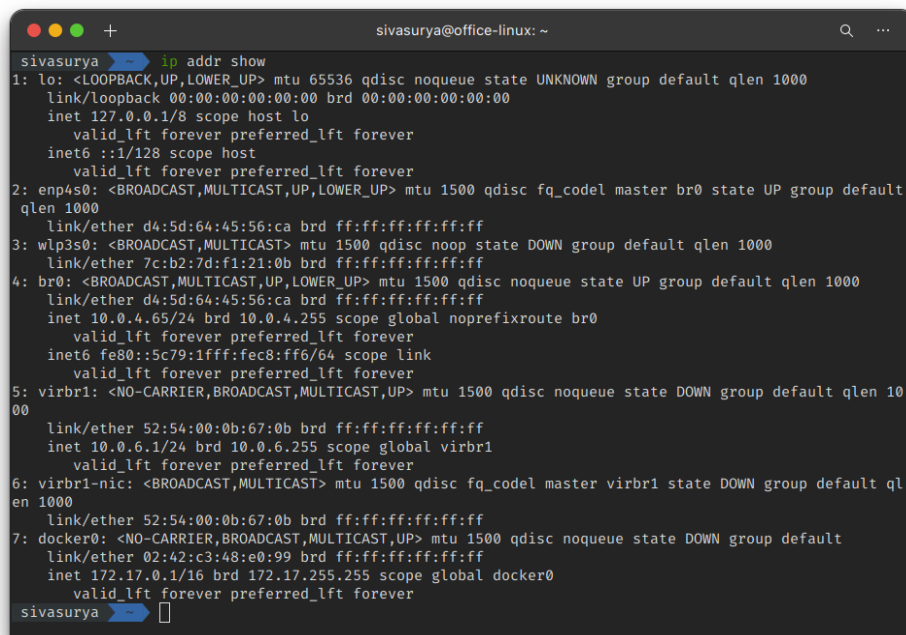
```
sivasurya@office-linux: ~  
sivasurya ~ sudo shutdown now
```

3. Date - displays the system date and time



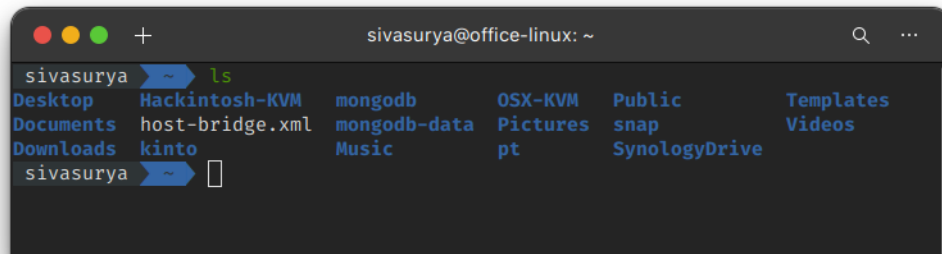
```
sivasurya ~ date
Sunday 06 September 2020 06:16:16 PM IST
sivasurya ~
```

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



```
sivasurya ~ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp4s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel master br0 state UP group default qlen 1000
   link/ether d4:5d:64:45:56:ca brd ff:ff:ff:ff:ff:ff
3: wlp3s0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
   link/ether 7c:b2:7d:f1:21:0b brd ff:ff:ff:ff:ff:ff
4: br0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
   link/ether d4:5d:64:45:56:ca brd ff:ff:ff:ff:ff:ff
   inet 10.0.4.65/24 brd 10.0.4.255 scope global noprefixroute br0
       valid_lft forever preferred_lft forever
   inet6 fe80::5c79:1fff:fec8:ff6/64 scope link
       valid_lft forever preferred_lft forever
5: virbr1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
   link/ether 52:54:00:0b:67:0b brd ff:ff:ff:ff:ff:ff
   inet 10.0.6.1/24 brd 10.0.6.255 scope global virbr1
       valid_lft forever preferred_lft forever
6: virbr1-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc fq_codel master virbr1 state DOWN group default qlen 1000
   link/ether 52:54:00:0b:67:0b brd ff:ff:ff:ff:ff:ff
7: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
   link/ether 02:42:c3:48:e0:99 brd ff:ff:ff:ff:ff:ff
   inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
sivasurya ~
```

5. Ls - lists directories



```
sivasurya ~ ls
Desktop  Hackintosh-KVM  mongodb  OSX-KVM  Public  Templates
Documents host-bridge.xml  mongodb-data  Pictures  snap  Videos
Downloads kinto           Music       pt        SynologyDrive
sivasurya ~
```

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

Client.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

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

Server.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