AMRITANAND
P2CSN17003

Assignment _3

1. What is the difference between a binary file and a text file?

Text files contain only textual data, binary files may contain both textual and custom binary data. bits in text files represent characters, while the bits in binary files represent custom data.

2. What is the ELF FILE Format? 2. What is a hexdump?

Executable and Linkable Format, An executable file using the ELF file format consists of an ELF header, followed by a program header table or a section header table, or both.

In computing, a hex dump is a hexadecimal view (on screen or paper) of computer data, from RAM or from a file or storage device. Looking at a hex dump of data is commonly done as a part of debugging, or of reverse engineering. In a hex dump, each byte (8-bits) is represented as a two-digit hexadecimal number.

3. What kind of information is stored in a hexdump?

A hex dump is a representation of a binary data stream where the contents of that stream are displayed as hexadecimal values.

4. Diagrammatically represent how the hex characters “DEADBEEF” will be represented in Big Endian and Little Endian formats.

Little Endian

D	E	Li A	D	B	E	E	F
1000	1001	1002	1003	1004	1005	1006	1007

D	E	A	D	B	E	E	F
1007	1006	1005	1004	1003	1002	1001	1000

Big endian

Part B

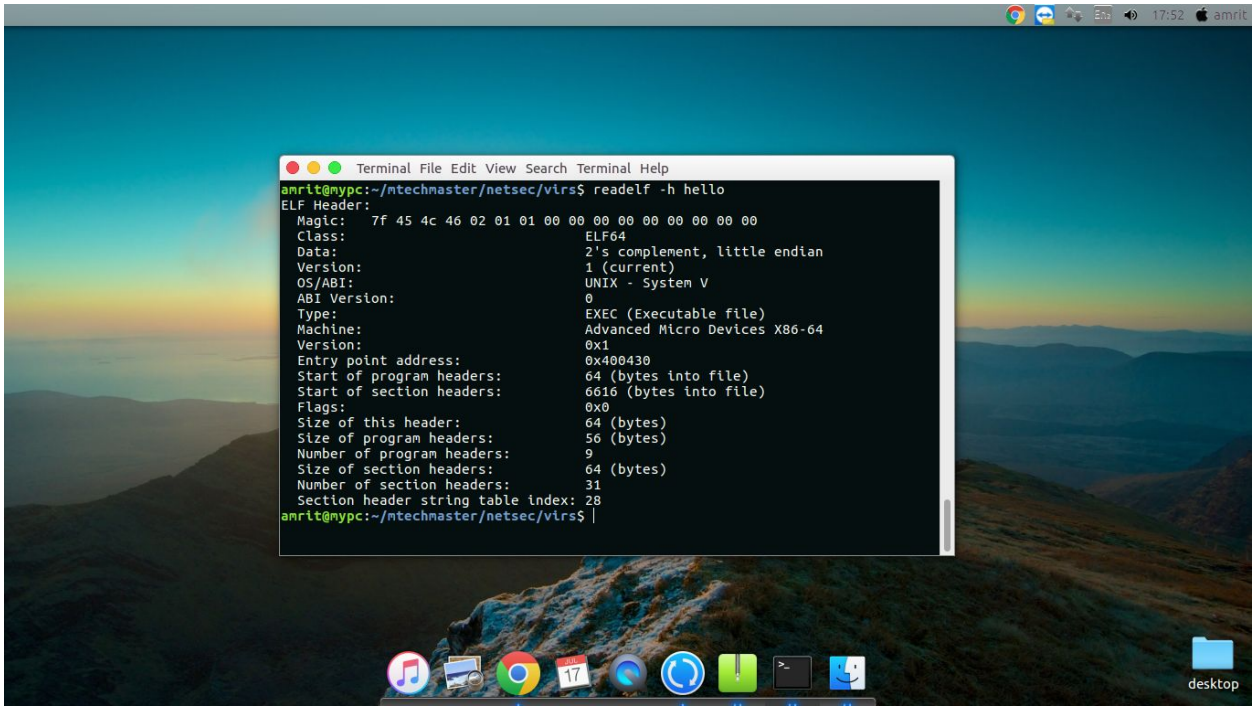
Q1

AMRITANAND
P2CSN17003

Assignment_3

The size in number of bytes of the hello world program is
8600 bytes

Q2
readelf -h hello

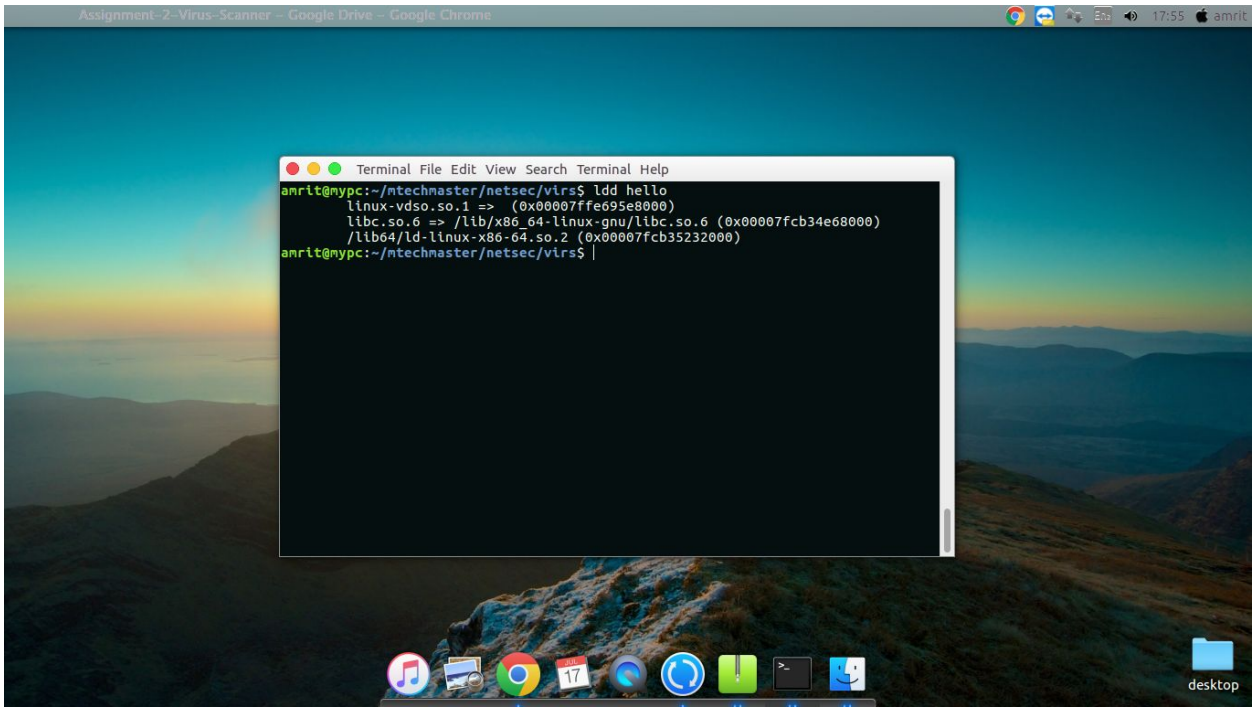


Q3

AMRITANAND
P2CSN17003

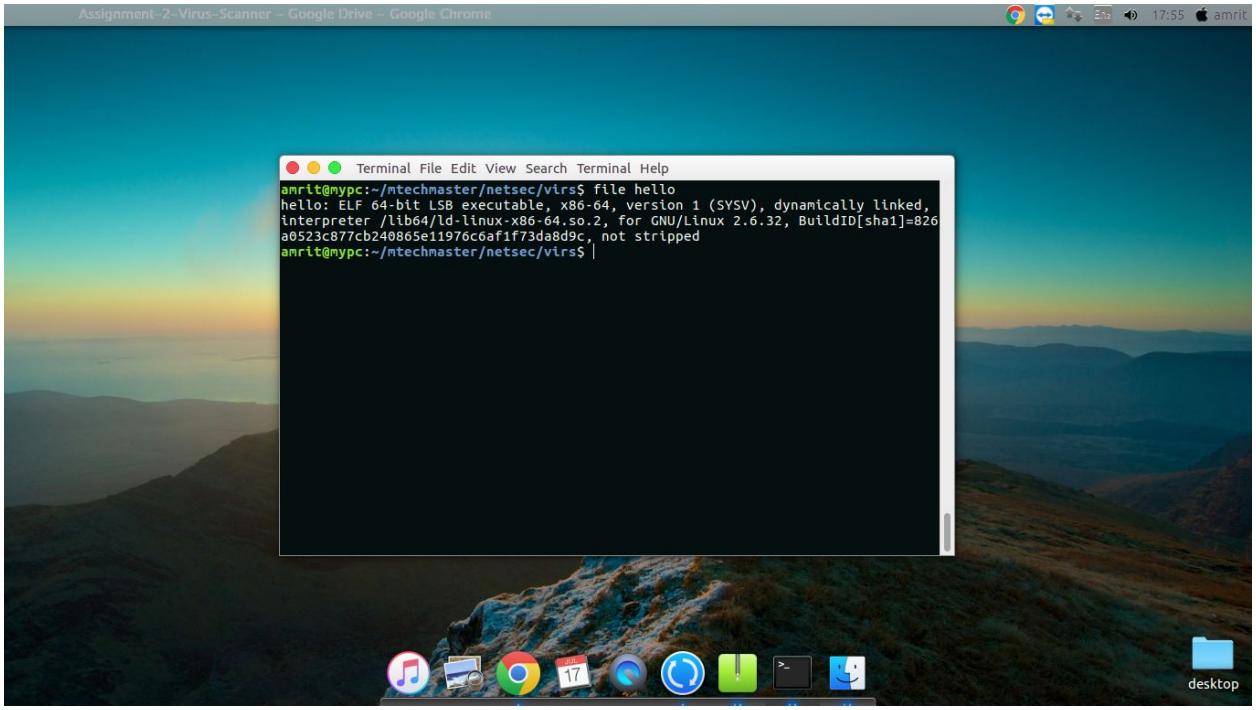
Assignment_3

ldd hello



Q4

file hello



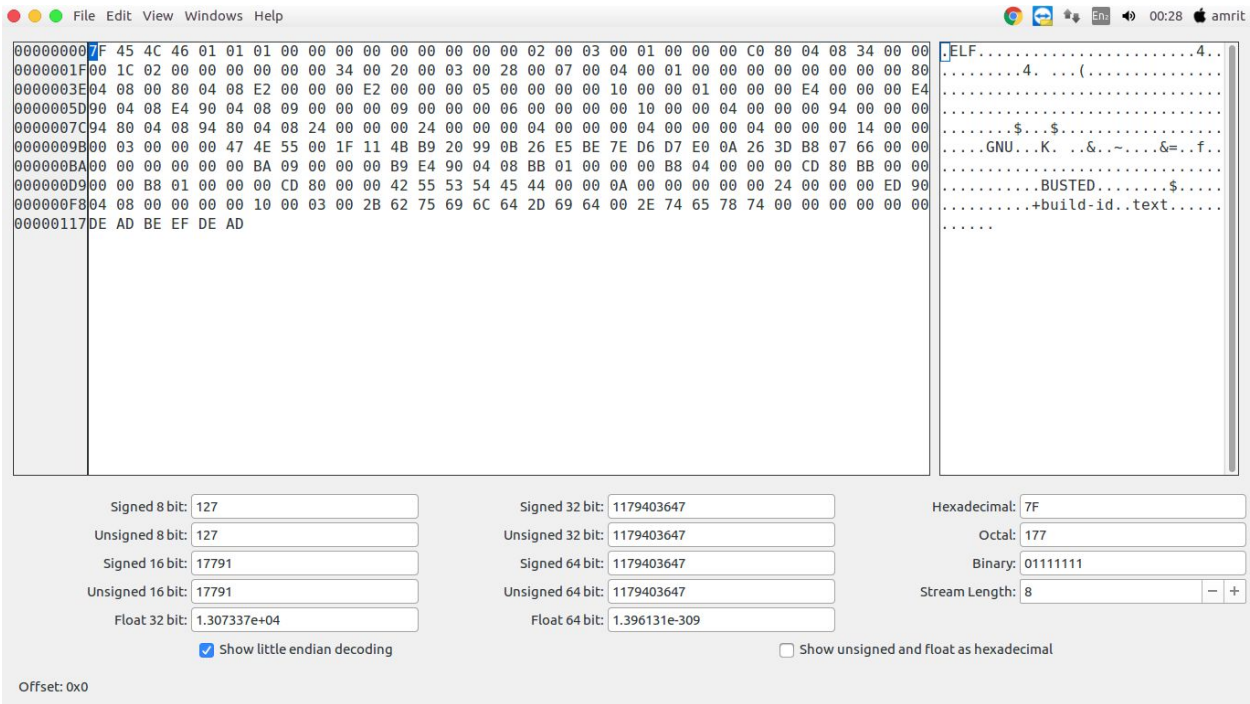
Q5

strip –s hello

It removes the null symbols from the executables.

AMRITANAND
P2CSN17003

Assignment_3

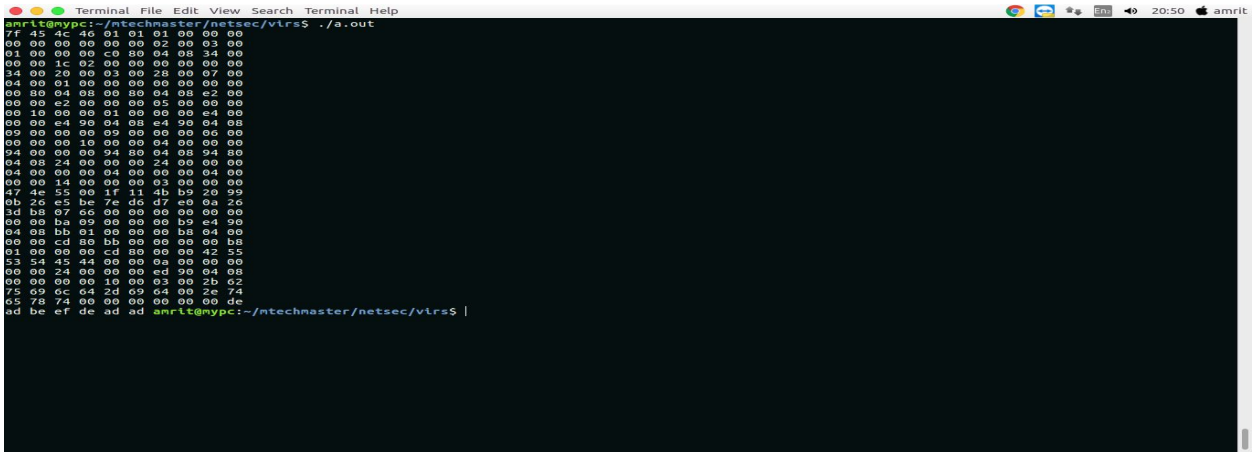


Q9.

```
#include<stdio.h>
#include<stdlib.h>
int main(){
unsigned char buffer[10];
FILE *ptr;

ptr = fopen("hello","rb");  // r for read, b for
binary
while(fread(buffer,sizeof(buffer),1,ptr)){
for(int i = 0; i<10; i=i+2)
   {
   printf("%02x %02x", buffer[i],buffer[i+1]);
   printf(" ");

   }
   printf("\n");
}
fread(buffer,sizeof(buffer),1,ptr);
for(int i = 0; i<5; i=i+2)
   {int x=1;
   if(i==4)
   {x=0;}
   printf("%02x %02x", buffer[i],buffer[i+x]);
   printf(" ");

   }

}
```

AMRITANAND
P2CSN17003

Assignment_3



Q11.

```
#include <stdio.h>
#include <dirent.h>
#include<string.h>
#include <sys/types.h>
#include <sys/stat.h>
#include<stdlib.h>

int main(void)
{
   struct dirent *de;  // Pointer for directory entry
   char filename[10];
   // opendir() returns a pointer of DIR type.
   DIR *dr = opendir(".");
   if (dr == NULL)  // opendir returns NULL if couldn't open directory
   {
      printf("Could not open current directory" );
      return 0;
   }
   // Refer http://pubs.opengroup.org/onlinepubs/7990989775/xsh/readdir.html
    FILE *ptr,*pt,*p;
    p=fopen("hello","rb");
    char helloco[300];
    fread(helloco,300,1,p);
    char buff[4];
   // for readdir()
   while ((de = readdir(dr)) != NULL){
         // printf("%s\n", de->d_name);
            strcpy(filename, de->d_name);


ptr=fopen(filename,"rb");

//fread(buff,4,1,ptr);
            // printf("%s\n", filename);

//printf("%s\n",&(filename[strlen(filename)-4]));
            if( strcmp(&(filename[strlen(filename)-4]),".bin")==0)
            {

pt=fopen(filename,"wb");
```

AMRITANAND
P2CSN17003

Assignment_3

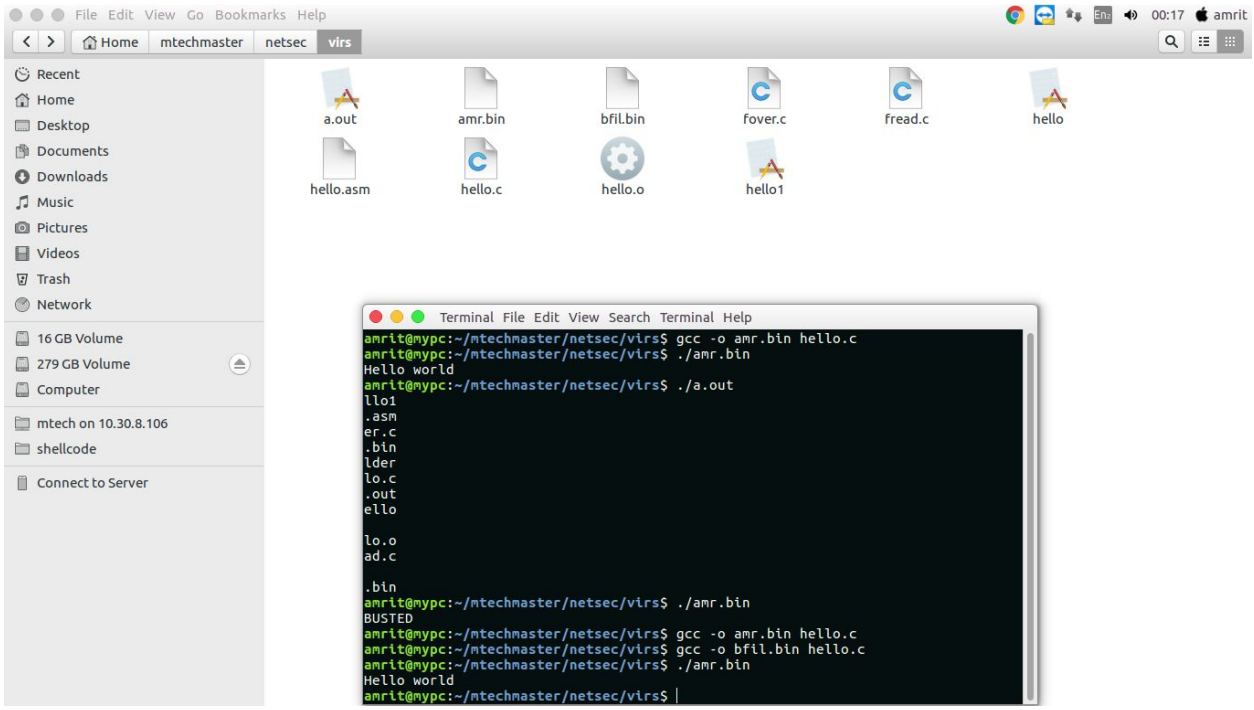
```
fwrite(helloco,300,1,pt);
        fclose(pt);

        }
        fclose(ptr);

}   closedir(dr);
fclose(p);
```

```
/*
char *fd = "hello.asm";
struct stat *buf,*mystat;
char mybu[25];
buf = malloc(sizeof(struct
stat));

stat(fd, buf);
stat(mybu, &mystat);
```

```
int size = buf->st_size;
printf("%s",mybu);

free(buf);*/

    return 0;
}
```

AMRITANAND
P2CSN17003

Assignment_3



+