**OS Lab**

**Session 2 – Lab 4 (11/12/2020)**

Parthivi Choubey CSE – B - 5th semester

180905456 Roll. no. - 60

**Question 1**

**Code**

#include <stdlib.h>

#include <stdio.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <sys/wait.h>

#include <unistd.h>

int main()

{

printf("Enter file name: ");

char input[256];

scanf("%s",input);

struct stat sbuf;

int ret = stat(input, &sbuf);

if(ret)

{

perror("stat");

return 1;

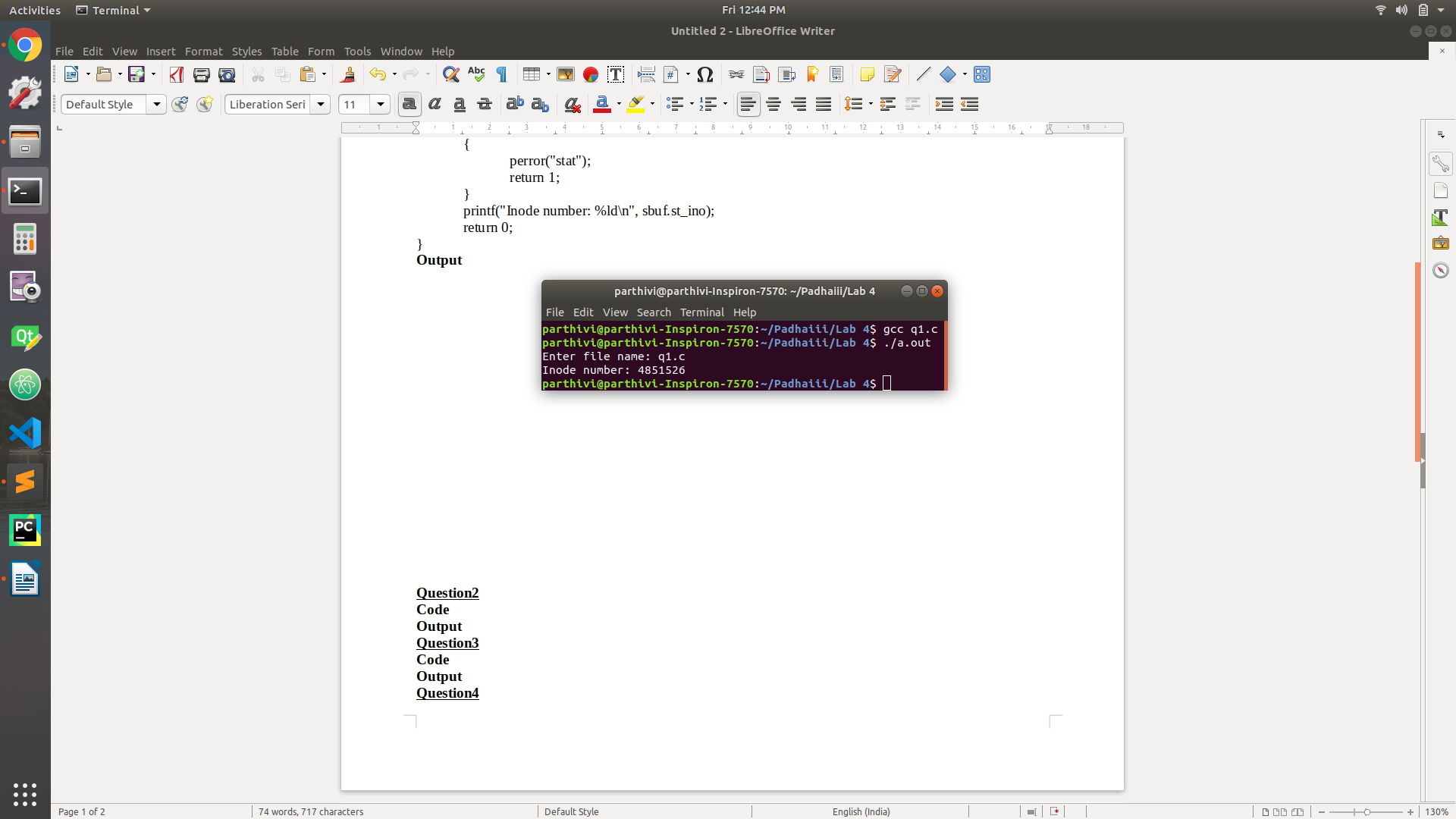
}

printf("Inode number: %ld\n", sbuf.st\_ino);

return 0;

}

**Output**

****

**Question 2**

**Code**

#include <stdlib.h>

#include <stdio.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <sys/wait.h>

#include <unistd.h>

int main()

{

printf("Enter file name: ");

char input[256];

scanf("%s",input);

struct stat sbuf;

int n = stat(input, &sbuf);

if(n)

{

perror("stat");

return 1;

}

printf("ID of device containing file: %ld\n", sbuf.st\_dev);

printf("Inode no.: %ld\n", sbuf.st\_ino);

printf("No. of hard links: %ld\n", sbuf.st\_nlink);

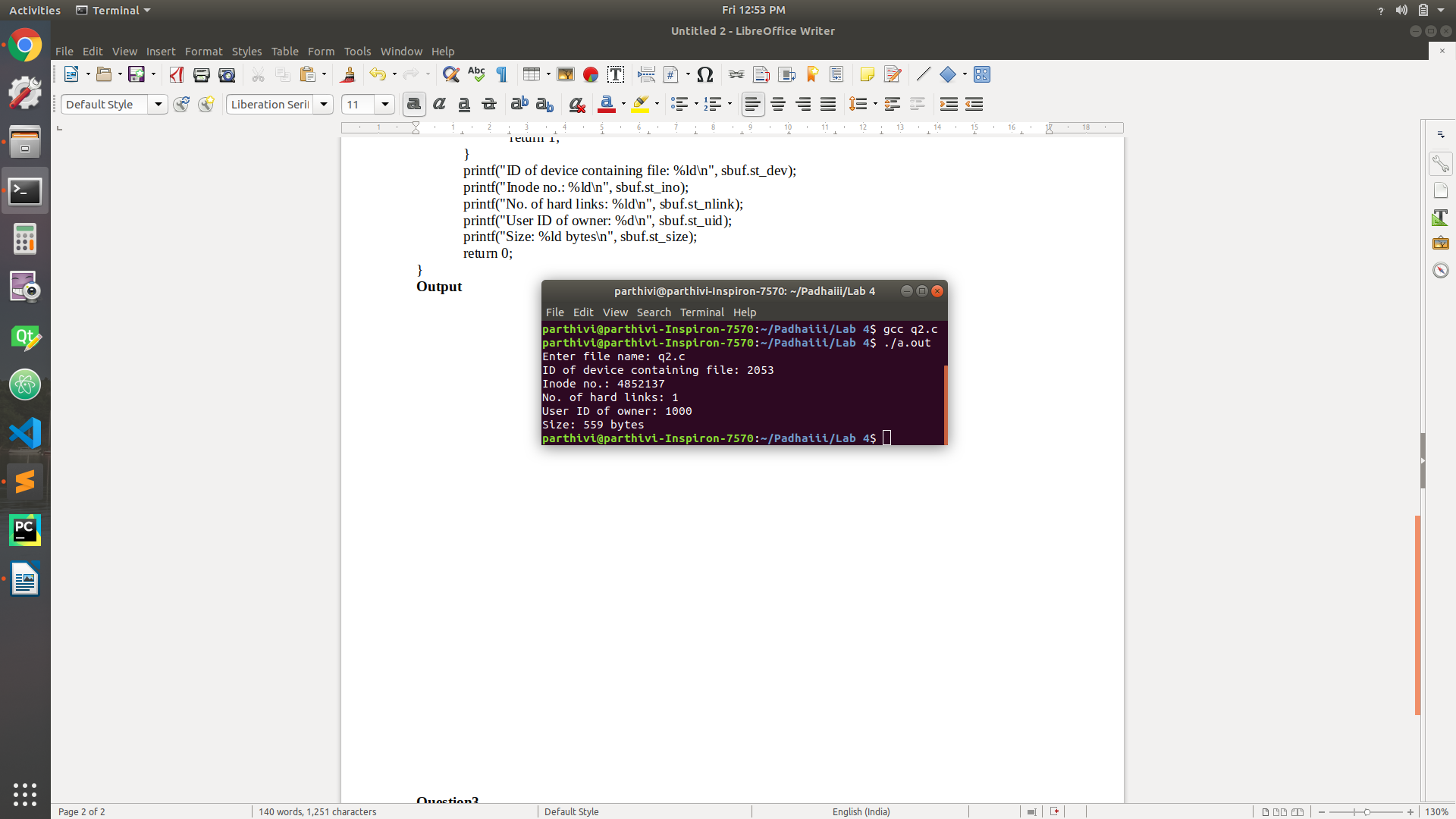
printf("User ID of owner: %d\n", sbuf.st\_uid);

printf("Size: %ld bytes\n", sbuf.st\_size);

return 0;

}

**Output**

****

**Question 3**

**Code**

#include <stdlib.h>

#include <stdio.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <sys/wait.h>

#include <unistd.h>

int main(int argc, char\* argv[])

{

if(argc != 2)

{

printf("Invalid number of arguments\n");

exit(-1);

}

int ret = link(argv[1], "./hardlink\_new");

if(ret)

{

perror("link");

return 1;

}

else

{

ret = unlink(argv[1]);

if(ret)

{

perror("unlink");

return 2;

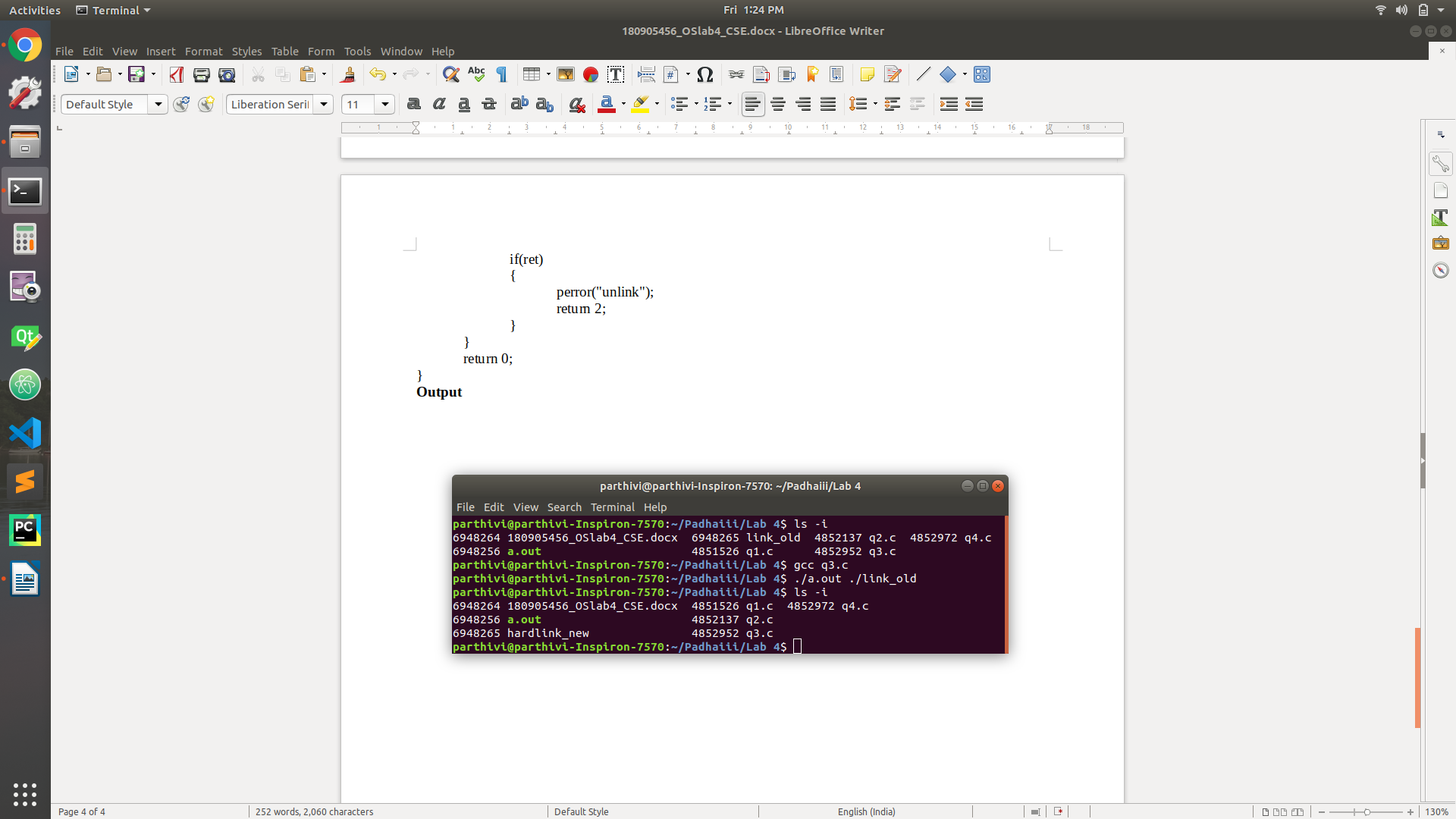
}

}

return 0;

}

**Output**

****

**Question 4**

**Code**

#include <stdlib.h>

#include <stdio.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <sys/wait.h>

#include <unistd.h>

int main(int argc, char\* argv[])

{

if(argc != 2)

{

printf("Invalid number of arguments\n");

exit(-1);

}

int ret = symlink(argv[1], "./symlink\_new");

if(ret)

{

perror("symlink");

return 1;

}

else

{

ret = unlink(argv[1]);

if(ret)

{

perror("unlink");

return 2;

}

}

return 0;

}

**Output**

