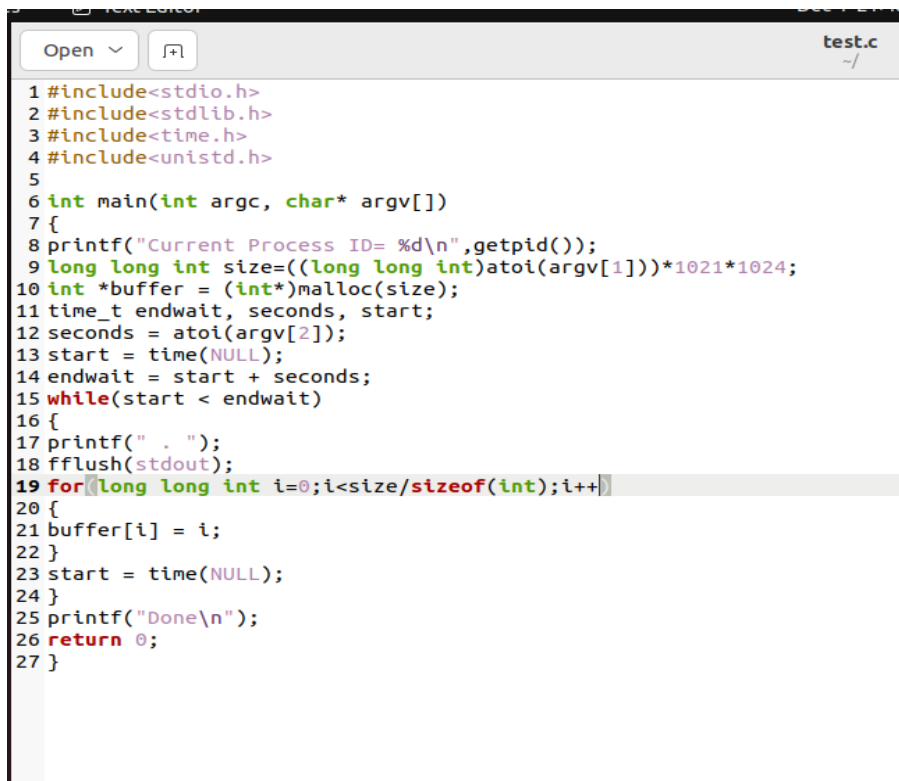


Experiment – 6

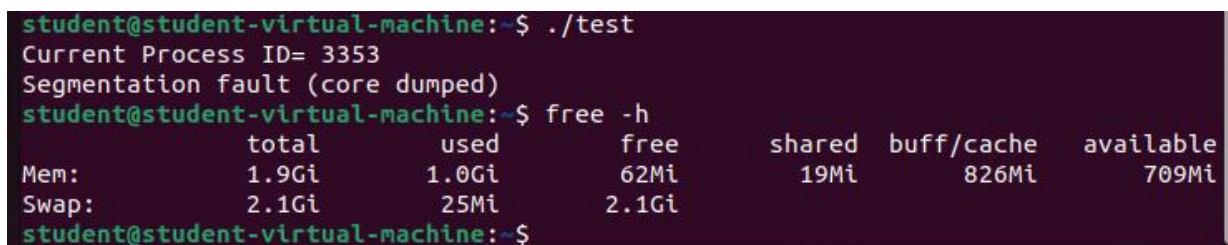
Write a C program that takes, as a command line argument, the number of megabytes of memory it will use and during execution it should consume that much memory. Observe memory usage during program execution using free command.

C Program:

A screenshot of a text editor window titled 'test.c' showing a C program. The program includes headers for stdio, stdlib, time, and unistd. It defines a main function that takes argc and argv. It prints the current process ID, calculates a size based on argv[1], allocates a buffer of that size, and enters a while loop that prints dots and flushes stdout until start + seconds. It then prints 'Done' and returns 0.

```
1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<time.h>
4 #include<unistd.h>
5
6 int main(int argc, char* argv[])
7 {
8     printf("Current Process ID= %d\n",getpid());
9     long long int size=((long long int)atoi(argv[1]))*1021*1024;
10    int *buffer = (int*)malloc(size);
11    time_t endwait, seconds, start;
12    seconds = atoi(argv[2]);
13    start = time(NULL);
14    endwait = start + seconds;
15    while(start < endwait)
16    {
17        printf(" . ");
18        fflush(stdout);
19        for(long long int i=0;i<size/sizeof(int);i++)
20        {
21            buffer[i] = i;
22        }
23        start = time(NULL);
24    }
25    printf("Done\n");
26    return 0;
27 }
```

Output:

A screenshot of a terminal window showing the execution of the program. The user runs './test' and the program prints the process ID and then a segmentation fault. The user then runs 'free -h' to show memory usage, which displays a table of memory statistics.

```
student@student-virtual-machine:~$ ./test
Current Process ID= 3353
Segmentation fault (core dumped)
student@student-virtual-machine:~$ free -h
```

	total	used	free	shared	buff/cache	available
Mem:	1.9Gi	1.0Gi	62Mi	19Mi	826Mi	709Mi
Swap:	2.1Gi	25Mi	2.1Gi			

```
student@student-virtual-machine:~$
```