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:

```
  GNU nano 6.2                                                  program.c *
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
#include<unistd.h>

int main(int argc, char* argv[]){
    printf("Current Process ID = %d\n",getpid());
    long long int size = ((long long int)atoi(argv[1]))*1024*1024; //in bytes
    int* buffer = (int*)malloc(size);

    //run the while loop for given amount of time.
    time_t endwait, seconds, start;
    seconds=atoi(argv[2]);
    start=time(NULL);
    endwait = start + seconds;
    while(start<endwait){
        printf(".");
        fflush(stdout);
        for(long long int i=0; i<size/sizeof(int); i++){
            buffer[i] = i;
        }

        start = time(NULL);
    }
    printf("(done)\n");
    return 0;

}
```

```
Every 1.0s: free -m

               total        used        free      shared  buff/cache   available
Mem:            3414         813        1699          46         900        2411
Swap:           4507           0        4507
```

```
rohin@rohin-VirtualBox:~/new/exp6$ gcc program.c
rohin@rohin-VirtualBox:~/new/exp6$ ./a.out
Current Process ID = 2667
Segmentation fault (core dumped)
rohin@rohin-VirtualBox:~/new/exp6$ watch -n 1 free -m

[1]+  Stopped                 watch -n 1 free -m
rohin@rohin-VirtualBox:~/new/exp6$ ./a.out 1000 20
Current Process ID = 2739
........(done)
rohin@rohin-VirtualBox:~/new/exp6$ watch -n 1 free -m
rohin@rohin-VirtualBox:~/new/exp6$
```

```
Every 1.0s: free -m

               total        used        free      shared  buff/cache   available
Mem:            3414         847        1665          46         901        2377
Swap:           4507           0        4507
```