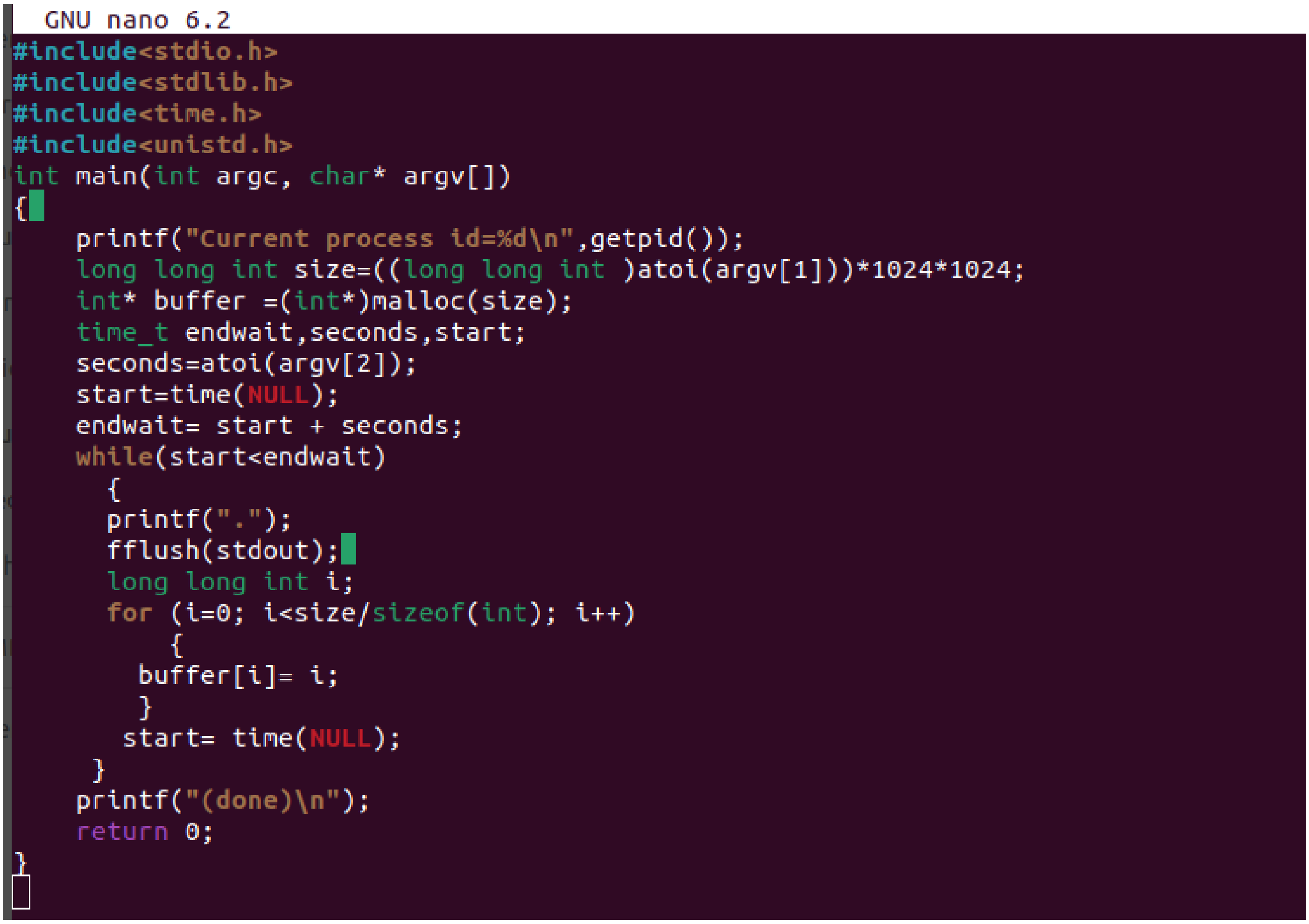
EXPERIMENT 6:

Title: 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.

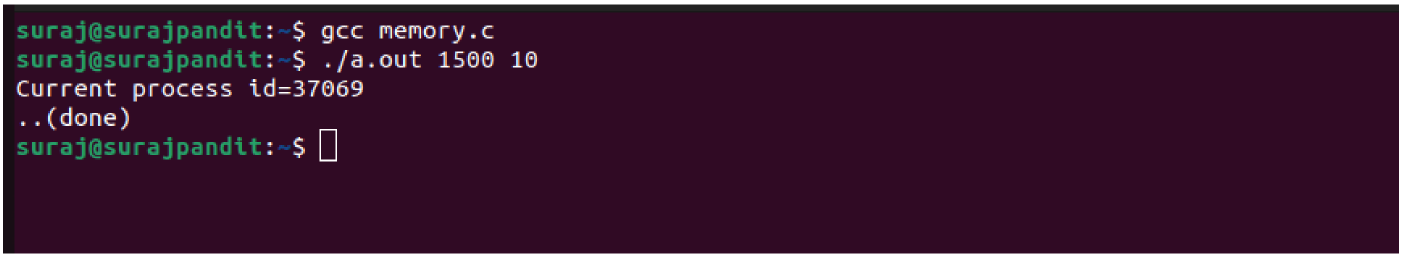
```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
#include<unistd.h>
int main(int argc, char* argv[])
{
printf("Current process id=%d\n",getpid());
long long int size=((long long int )atoi(argv[1]))*1024*1024;
int* buffer =(int*)malloc(size);
time_t endwait,seconds,start;
seconds=atoi(argv[2]);
start=time(NULL);
endwait= start + seconds;
while(start<endwait)
{
printf(".");
fflush(stdout);
long long int i;
for (i=0; i<size/sizeof(int); i++)
{
buffer[i]= i;
}
start= time(NULL);
}
printf("(done)\n");
return 0;
}
```

CODE:

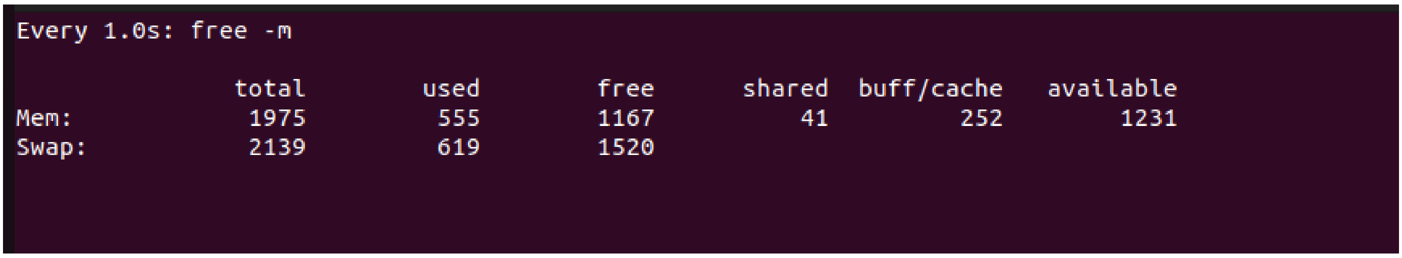


```
  GNU nano 6.2
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
#include<unistd.h>
int main(int argc, char* argv[])
{
    printf("Current process id=%d\n",getpid());
    long long int size=((long long int )atoi(argv[1]))*1024*1024;
    int* buffer =(int*)malloc(size);
    time_t endwait,seconds,start;
    seconds=atoi(argv[2]);
    start=time(NULL);
    endwait= start + seconds;
    while(start<endwait)
      {
      printf(".");
      fflush(stdout);
      long long int i;
      for (i=0; i<size/sizeof(int); i++)
          {
        buffer[i]= i;
        }
       start= time(NULL);
     }
    printf("(done)\n");
    return 0;
}
```

OUTPUT:



```
suraj@surajpandit:~$ gcc memory.c
suraj@surajpandit:~$ ./a.out 1500 10
Current process id=37069
..(done)
suraj@surajpandit:~$ 
```



```
Every 1.0s: free -m

               total        used        free      shared  buff/cache   available
Mem:            1975         555        1167          41         252        1231
Swap:           2139         619        1520
```