

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



The screenshot shows a terminal window with the title bar "rakshit@RG: ~/sample". The terminal is running the GNU nano 6.2 editor, editing a file named "exp6.c". The code in the editor is as follows:

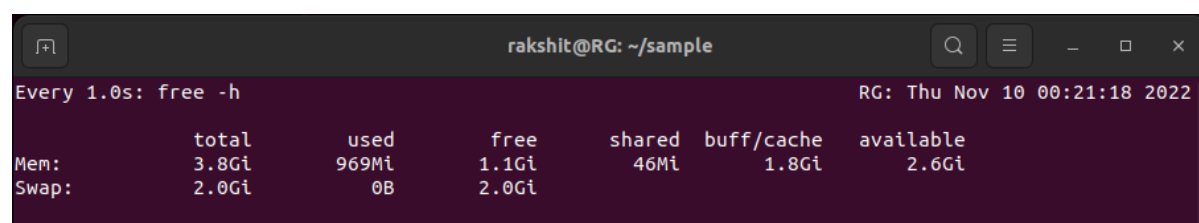
```
GNU nano 6.2 exp6.c *
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
#include<unistd.h>

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;
}
```

→Before executing the program



The screenshot shows a terminal window with the title bar "rakshit@RG: ~/sample". The terminal is running the command "free -h". The output is as follows:

```
Every 1.0s: free -h RG: Thu Nov 10 00:21:18 2022
```

	total	used	free	shared	buff/cache	available
Mem:	3.8Gi	969Mi	1.1Gi	46Mi	1.8Gi	2.6Gi
Swap:	2.0Gi	0B	2.0Gi			

→After execution

```
rakshit@RG: ~/sample
rakshit@RG:~/sample$ ./a.out 1000 20
Current Process ID = 6427
.....(done)
rakshit@RG:~/sample$
```

```
rakshit@RG: ~/sample
Every 1.0s: free -h                                     RG: Thu Nov 10 00:21:18 2022
```

	total	used	free	shared	buff/cache	available
Mem:	3.8Gi	969Mi	1.1Gi	46Mi	1.8Gi	2.6Gi
Swap:	2.0Gi	0B	2.0Gi			