

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.

Syntax :

```
#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 int size= ((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);
        for(long int i=0; i<size/sizeof(int); i++)
        {
            buffer[i] = i;
        }
        Start= time(NULL);
    }
    printf("(done)\n");
    return 0;
}
```

```

#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;
}

```

```

sudanshi@sudanshi:~/Desktop$ free -m

```

	total	used	free	shared	buff/cache	available
Mem:	10739	838	9079	37	821	9621
Swap:	2047	0	2047			

```

sudanshi@sudanshi:~/Desktop$ ./a.out 800 20
Current Process ID = 2632
.....(done)
sudanshi@sudanshi:~/Desktop$ free -m

```

	total	used	free	shared	buff/cache	available
Mem:	10739	774	9085	39	879	9682
Swap:	2047	0	2047			

```

sudanshi@sudanshi:~/Desktop$ █

```