

SHARED MEMORY - STATIC

s[LEN] shared

```
func<<< block, thread >>>();
```

```
#define LEN 1000
 2
 3
     __global__ void gpu_func(int *arr, int sz) {
         __shared__ int s[LEN];
 5
         int id = threadId.x;
 6
         int bs = blockDim.x;
 7
         for (int i = 0; i < LEN / bs; i++) {
 8
              s[i * bs + id] = arr[i * bs + id];
 9
         __syncthreads();
10
11
12
13
     int main(int argc, char *argv[]) {
14
15
         gpu_func<<<1, 100 >>>(gpu_arr, LEN);
16
         return 0;
```

SHARED MEMORY - DYNAMIC

shared extern

```
func<<< block, thread, SM_size >>>();
```

```
#define LEN 1000
 3
     __global__ void gpu_func(int *arr, int sz) {
         extern __shared__ int s[];
 5
         int id = threadId.x;
 6
         int bs = blockDim.x;
         for (int i = 0; i < LEN / bs; i++) {
 8
             s[i * bs + id] = arr[i * bs + id];
 9
10
         __syncthreads();
11
12
13
14
     int main(int argc, char *argv[]) {
         gpu_func<<< 10, 100, sizeof(int)*LEN >>>(gpu_arr, LEN);
15
16
         return 0;
17
```