SITCON'21

MEMORY LOCATION

- CPU：RAM
- GPU：Global memory

- Malloc :

```
1    int cpu_arr[LEN];
2    int *gpu_arr;
3    cudaMalloc(&gpu_arr, sizeof(int) * LEN);
```

- Memcpy :

```
4    cudaMemcpy(gpu_arr, cpu_arr, sizeof(int) * LEN, cudaMemcpyDeviceToHost);
5    cudaMemcpy(cpu_arr, gpu_arr, sizeof(int) * LEN, cudaMemcpyHostToDevice);
```

```
cudaMemcpyHostToDevice);
```

```
cudaMemcpyDeviceToHost);
```

MEMORY LOCATION

- CPU：RAM
- GPU：Global memory

- Malloc :

```
1   int cpu_arr[LEN];
2   int *gpu_arr;
3   cudaMalloc(&gpu_arr, sizeof(int) * LEN);
```

- Memcpy :

```
4   cudaMemcpy(gpu_arr, cpu_arr, sizeof(int) * LEN, cudaMemcpyHostToDevice);
5   cudaMemcpy(cpu_arr, gpu_arr, sizeof(int) * LEN, cudaMemcpyDeviceToHost);
```

DECLARE & EXCUTE FUNCTION

- `__global__ void name(…);`
- `name<<<grid_size, block_size>>>(…);`

```
#define LEN 1000
__global__ void gpu_func(int add, int *arr) {
    arr[0] += add;
}

int main(int argc, char *argv[]) {
    int cpu_arr[LEN];
    int *gpu_arr;
    cudaMalloc(&gpu_arr, sizeof(int) * LEN);
    cudaMemcpy(gpu_arr, cpu_arr, sizeof(int) * LEN, cudaMemcpyHostToDevice);
    gpu_func<<<10, 100>>>(87, gpu_arr);
    cudaMemcpy(cpu_arr, gpu_arr, sizeof(int) * LEN, cudaMemcpyDeviceToHost);
}

```