

# به نام خدا

علیرضا آخوندی  
۹۷۳۱۱۰۷

گام اول :  
خروجی حاصل از گام اول :

```
(base) ~alirezaak@ALIPC ~/multicore programming/lab6
└─$ ./matmul
[Matrix Multiply Using CUDA] - Starting...
GPU Device 0: "NVIDIA GeForce GTX 1050" with compute capability 6.1

[-] N = 32
MatrixA(32,32), MatrixB(32,32)
Computing result using CUDA Kernel...
Elapsed time in msec = 0.033824
```

گام دوم و سوم :  
زمان اجرای برای راه های مختلف (matmul3v2 همان روش سوم است که با کمک shared memory پیاده سازی شده است)

```
(base) ~alirezaak@ALIPC ~/multicore programming/lab6
└─$ ./matmul3v2
[Matrix Multiply Using CUDA] - Starting...
GPU Device 0: "NVIDIA GeForce GTX 1050" with compute capability 6.1

MatrixA(2048,2048), MatrixB(2048,2048)
Computing result using CUDA Kernel...
Elapsed time in msec = 143.195801
(base) ~alirezaak@ALIPC ~/multicore programming/lab6
└─$ ./matmul3
[Matrix Multiply Using CUDA] - Starting...
GPU Device 0: "NVIDIA GeForce GTX 1050" with compute capability 6.1

MatrixA(2048,2048), MatrixB(2048,2048)
Computing result using CUDA Kernel...
Elapsed time in msec = 1274.613037
(base) ~alirezaak@ALIPC ~/multicore programming/lab6
└─$ ./matmul2
[Matrix Multiply Using CUDA] - Starting...
GPU Device 0: "NVIDIA GeForce GTX 1050" with compute capability 6.1

MatrixA(2048,2048), MatrixB(2048,2048)
Computing result using CUDA Kernel...
Elapsed time in msec = 202.531067
(base) ~alirezaak@ALIPC ~/multicore programming/lab6
└─$ ./matmul1
[Matrix Multiply Using CUDA] - Starting...
GPU Device 0: "NVIDIA GeForce GTX 1050" with compute capability 6.1

MatrixA(2048,2048), MatrixB(2048,2048)
Computing result using CUDA Kernel...
Elapsed time in msec = 3593.869385
```

همانطور که دیده می شود روش **tiling** ای که در راه سوم استفاده شده زمان اجرای بدتری نسبت به راه دوم دارد اما با بهینه کردن دسترسی به حافظه و خود داری از **access** های اضافه ، تسریع بیشتری با تعداد نخ کمتر نسبت به روش دوم گرفتیم.