

Parallel Computing Workshop

Ms. Kranti Ingale

Prime Minister's Research Fellows (**PMRF**) Scholar

PACE Lab - CSE, IIT Madras

Roadmap

- Introduction to GPU
- CUDA Program Flow and CPU-GPU Communication
- Thread organization (Grids, Blocks, Threads, 1D/2D)
- CUDA Memory Model
- CUDA Functions
- CUDA Thrust

Roadmap

- **Introduction to GPU**
- CUDA Program Flow and CPU-GPU Communication
- Thread organization (Grids, Blocks, Threads, 1D/2D)
- CUDA Memory Model
- CUDA Functions
- CUDA Thrust

Hello World

```
#include <stdio.h>

int main() {
    printf("Hello World.\n");
    return 0;
}
```

Hello World

```
#include <stdio.h>
int main() {
    printf("Hello World.\n");
    return 0;
}
```

Compile: nvcc <ProgramName>.cu
Run: a.out

GPU Hello World program

```
#include <stdio.h>
#include<cuda.h>

__global__ void Mykernel() {
    printf("Hello World.\n");
}

int main() {
    Mykernel<<<1,1>>>();
    return 0;
}
```

GPU Hello World program

```
#include <stdio.h>
#include<cuda.h>

__global__ void Mykernel() {
    printf("Hello World.\n");
}

int main() {
    Mykernel<<<1,1>>>();
    return 0;
}
```

GPU Hello World program

CUDA Header

```
#include <stdio.h>
#include<cuda.h>

__global__ void Mykernel() {
    printf("Hello World.\n");
}

int main() {
    Mykernel<<<1,1>>>();
    return 0;
}
```

Note :

cuda.h is header file needs to include in order to run CUDA programs

GPU Hello World program

Kernel

```
#include <stdio.h>
#include<cuda.h>

__global__ void Mykernel() {
    printf("Hello World.\n");
}

int main() {
    Mykernel<<<1,1>>>();
    return 0;
}
```

Note :

Kernel is Function that :

- Runs on the device
- Is called from host code

GPU Hello World program

```
#include <stdio.h>
#include<cuda.h>

__global__ void Mykernel() {
    printf("Hello World.\n");
}

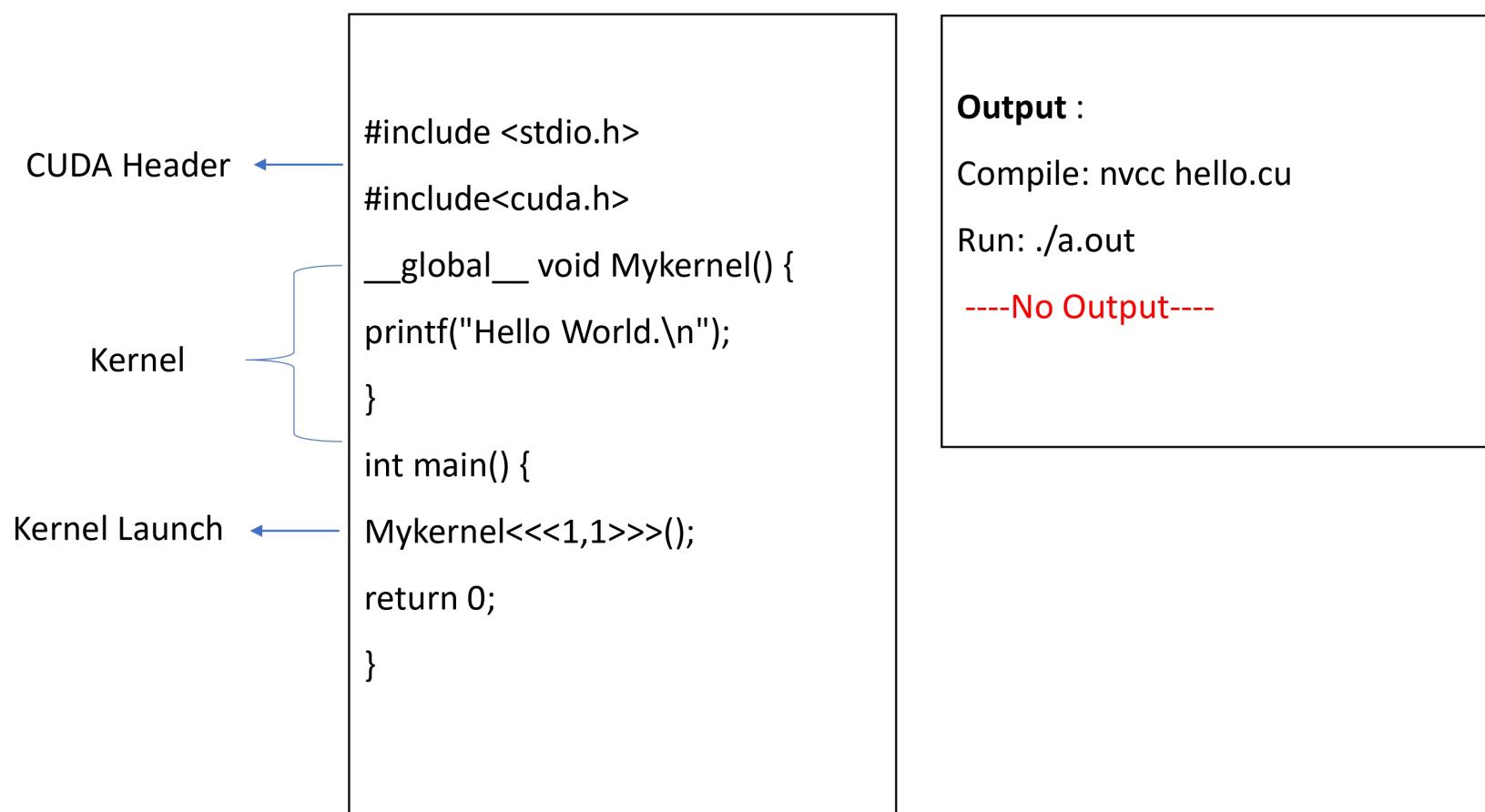
int main() {
    Mykernel<<<1,1>>>();
    return 0;
}
```

Kernel Launch ←

Note :

Kernel are executed N times in parallel by N different *CUDA threads*.

GPU Hello World program



GPU Hello World program

```
#include <stdio.h>
#include<cuda.h>

__global__ void Mykernel() {
    printf("Hello World\n");
}

int main() {
    Mykernel<<<1,1>>>();
    cudaDeviceSynchronize();
    return 0;
}
```

Output :

Hello World

More GPU Hello World program

```
#include <stdio.h>
#include<cuda.h>

__global__ void Mykernel() {
    printf("Hello World\n");
}

int main() {
    Mykernel<<<1,1>>>();
    Mykernel<<<1,1>>>();
    cudaDeviceSynchronize();
    return 0;
}
```

More GPU Hello World program

```
#include <stdio.h>
#include<cuda.h>

__global__ void Mykernel() {
    printf("Hello World\n");
}

int main() {
    Mykernel<<<1,1>>>();
    Mykernel<<<1,1>>>();
    cudaDeviceSynchronize();
    return 0;
}
```

Output :

Hello World
Hello World

More GPU Hello World program

```
#include <stdio.h>
#include<cuda.h>

__global__ void Mykernel() {
    printf("Hello World\n");
}

int main() {
    Mykernel<<<1,1>>>();
    cudaDeviceSynchronize();
    Mykernel<<<1,1>>>();
    return 0;
}
```

More GPU Hello World program

```
#include <stdio.h>
#include<cuda.h>

__global__ void Mykernel() {
    printf("Hello World\n");
}

int main() {
    Mykernel<<<1,1>>>();
    cudaDeviceSynchronize();
    Mykernel<<<1,1>>>();
    return 0;
}
```

Output :

Hello World
Hello World (Optional)

More GPU Hello World program

```
__global__ void Mykernel1() {
    printf("Hello World 1\n");
}

__global__ void Mykernel2() {
    printf("Hello World 2\n");
}

int main() {
    Mykernel1<<<1,1>>>();
    Mykernel2<<<1,1>>>();
    cudaDeviceSynchronize();
    return 0;
}
```

More GPU Hello World program

```
__global__ void Mykernel1() {  
    printf("Hello World 1\n");  
}  
  
__global__ void Mykernel2() {  
    printf("Hello World 2\n");  
}  
  
int main() {  
    Mykernel1<<<1,1>>>();  
    Mykernel2<<<1,1>>>();  
    cudaDeviceSynchronize();  
    return 0; }
```

Output:

Hello World 1
Hello World 2

More GPU Hello World program

```
__global__ void Mykernel1() {
    printf("Hello World 1\n");
}

__global__ void Mykernel2() {
    printf("Hello World 2\n");
}

int main() {
    Mykernel1<<<1,1>>>();
    Mykernel2<<<1,1>>>();
    cudaDeviceSynchronize();
    printf("Inside main\n");
    return 0;
}
```

More GPU Hello World program

```
__global__ void Mykernel1() {  
    printf("Hello World 1\n");  
}  
  
__global__ void Mykernel2() {  
    printf("Hello World 2\n");  
}  
  
int main() {  
    Mykernel1<<<1,1>>>();  
    Mykernel2<<<1,1>>>();  
    cudaDeviceSynchronize();  
    printf("Inside main\n");  
    return 0; }
```

Output:

```
Hello World 1  
Hello World 2  
Inside main
```

More GPU Hello World program

```
__global__ void Mykernel1() {
    printf("Hello World 1\n");
}

__global__ void Mykernel2() {
    printf("Hello World 2\n");
}

int main() {
    Mykernel1<<<1,1>>>();
    Mykernel2<<<1,1>>>();
    printf("Inside main\n");
    cudaDeviceSynchronize();
    return 0;
}
```

More GPU Hello World program

```
__global__ void Mykernel1() {  
    printf("Hello World 1\n");  
}  
  
__global__ void Mykernel2() {  
    printf("Hello World 2\n");  
}  
  
int main() {  
    Mykernel1<<<1,1>>>();  
    Mykernel2<<<1,1>>>();  
    printf("Inside main\n");  
    cudaDeviceSynchronize();  
    return 0; }
```

Output :

Hello World 1
Hello World 2
Inside main

More GPU Hello World program

```
__global__ void Mykernel1() {  
    printf("Hello World 1\n");  
}  
  
__global__ void Mykernel2() {  
    printf("Hello World 2\n");  
}  
  
int main() {  
    Mykernel1<<<1,1>>>();  
    Mykernel2<<<1,1>>>();  
    printf("Inside main\n");  
    cudaDeviceSynchronize();  
    return 0; }
```

Output :

Hello World 1
Hello World 2
Inside main

Hello World 1
Inside main
Hello World 2

More GPU Hello World program

```
__global__ void Mykernel1() {  
    printf("Hello World 1\n");  
}  
  
__global__ void Mykernel2() {  
    printf("Hello World 2\n");  
}  
  
int main() {  
    Mykernel1<<<1,1>>>();  
    Mykernel2<<<1,1>>>();  
    printf("Inside main\n");  
    cudaDeviceSynchronize();  
    return 0; }
```

Output :

Hello World 1
Hello World 2
Inside main

Hello World 1
Inside main
Hello World 2

Inside main
Hello World 1
Hello World 2

More GPU Hello World program

```
__global__ void Mykernel() {  
    printf("Hello World \n");  
}  
  
int main() {  
    Mykernel<<<1,1>>>();  
    Printf("main one\n");  
    Mykernel<<<1,1>>>();  
    Printf("main two\n");  
    cudaDeviceSynchronize();  
    Printf("main three\n");  
    return 0; }
```

Identify which prints execute in parallel

More GPU Hello World program

```
#include <stdio.h>
#include<cuda.h>

__global__ void Mykernel() {
    printf("Hello World\n");
}

int main() {
    Mykernel<<<1,32>>>();
    return 0;
}
```

More GPU Hello World program

```
#include <stdio.h>
#include<cuda.h>
__global__ void Mykernel() {
    printf("Hello World\n");
}
int main() {
    Mykernel<<<1,32>>>();
    return 0;
}
```

Output :

Hello World
Hello World
.32 times

More GPU Hello World program

```
#include <stdio.h>
#include<cuda.h>
__global__ void Mykernel() {
printf("Hello World\n");
}
int main() {
Mykernel<<<x,y>>>();
return 0;
}
```

Output :

Hello World
Hello World
. .
x * y times

More GPU Hello World program

```
#include <stdio.h>
#include<cuda.h>
#define N 10
__global__ void Mykernel() {
    printf("%d\n",N);
}
int main() {
    Mykernel<<<x,y>>>();
    return 0;
}
```

More GPU Hello World program

```
#include <stdio.h>
#include<cuda.h>
#define N 10
__global__ void Mykernel() {
printf("%d\n",N);
}
int main() {
Mykernel<<<x,y>>>();
return 0;
}
```

Output :

10
10
. .
x * y times

GPU Hello World program

```
#include <stdio.h>
#include<cuda.h>
Const char *data = "Hello World";
__global__ void Mykernel() {
printf("%s\n",data);
}
int main() {
Mykernel<<<1,1>>>();
return 0;
}
```

GPU Hello World program

```
#include <stdio.h>
#include<cuda.h>
Const char *data = "Hello World";
__global__ void Mykernel() {
printf("%s\n",data);
}
int main() {
Mykernel<<<1,1>>>();
return 0;
}
```

Output :

ERROR

identifier "data" is undefined in
device code

Thank You