**PP Lab 7**

**Name:** GP Anirudh

**Registration Number:** 180905452

**Section:** B

**Batch:** B2

**Roll Number:** 59

1. Write a program in CUDA to count the number of times a given word is repeated in a sentence.

(Use Atomic function).

#include <cuda.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

\_\_global\_\_

void word\_count\_kernel(char\* str, char\* key, int\* word\_indices, int\* result)

{

int idx = threadIdx.x + blockIdx.x \* blockDim.x;

// get idx'th word

int si = word\_indices[idx];

int ei = word\_indices[idx+1];

char word[100];

int i = 0;

for (i = 0; i < (ei-si-1); i++)

{

word[i] = str[si+1+i];

}

word[i] = '\0';

// compare word and key

int i1 = 0;

int i2 = 0;

int is\_equal = 1;

while (word[i1] != '\0' && key[i2] != '\0')

{

if (word[i1] == key[i2])

{

i1++;

i2++;

}

else

{

is\_equal = 0;

break;

}

}

if (is\_equal == 1)

{

atomicAdd(result, 1);

}

}

int main()

{

char str[100] = "We are having PCAP Lab today morning";

char key[100] = "today";

int str\_len = strlen(str);

int key\_len = strlen(key);

int word\_count = 0;

for (int i = 0; i < str\_len; i++)

{

if (str[i] == ' ')

{

word\_count++;

}

}

word\_count--;

int\* word\_indices = (int\*) (malloc(word\_count \* sizeof(int)));

int wi = -1;

for (int i = 0; i < str\_len; i++)

{

if (str[i] == ' ')

{

word\_indices[++wi] = i;

}

}

int result = 0;

char\* d\_str;

char\* d\_key;

int\* d\_word\_indices;

int\* d\_result;

cudaMalloc((void\*\*)&d\_str, str\_len \* sizeof(char));

cudaMalloc((void\*\*)&d\_key, key\_len \* sizeof(char));

cudaMalloc((void\*\*)&d\_word\_indices, (word\_count+1) \* sizeof(int));

cudaMalloc((void\*\*)&d\_result, sizeof(int));

cudaMemcpy(d\_str, str, str\_len \* sizeof(char), cudaMemcpyHostToDevice);

cudaMemcpy(d\_key, key, key\_len \* sizeof(char), cudaMemcpyHostToDevice);

cudaMemcpy(d\_word\_indices, word\_indices, (word\_count+1) \* sizeof(int), cudaMemcpyHostToDevice);

cudaMemcpy(d\_result, &result, sizeof(int), cudaMemcpyHostToDevice);

word\_count\_kernel<<<1, word\_count>>>(d\_str, d\_key, d\_word\_indices, d\_result);

cudaMemcpy(&result, d\_result, sizeof(int), cudaMemcpyDeviceToHost);

printf("Input String: %s\n", str);

printf("Key: %s\n", key);

printf("Total occurances of %s is %d\n", key, result);

cudaFree(d\_str);

cudaFree(d\_key);

cudaFree(d\_result);

return 0;

}

**Output:**

****

2. Write a CUDA program that reads a string ***Sin*** and produces the string ***Sout*** as follows:

Input string ***Sin***: PCAP Output string ***Sout***: PCAP**PCA**PC**P**

#include <cuda.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

\_\_global\_\_

void kernel(char\* sin, int\* sin\_len, char\* sout)

{

int idx = threadIdx.x + blockIdx.x \* blockDim.x;

int si = 0; // start index

for (int i = 0; i < idx; i++)

{

si += (\*sin\_len)-i;

}

int total\_chars = (\*sin\_len) - idx;

for (int i = 0; i < total\_chars; i++)

{

sout[si++] = sin[i];

}

}

int main()

{

char sin[100] = "PCAP";

char sout[100];

int sin\_len = strlen(sin);

int sout\_len = 0;

for (int i = 0; i < sin\_len; i++)

{

sout\_len += (i+1);

}

char\* d\_sin;

int\* d\_sin\_len;

char\* d\_sout;

cudaMalloc((void\*\*) &d\_sin, sin\_len \* sizeof(char));

cudaMalloc((void\*\*) &d\_sin\_len, sizeof(int));

cudaMalloc((void\*\*) &d\_sout, (sout\_len + 1) \* sizeof(char));

cudaMemcpy(d\_sin, sin, sin\_len \* sizeof(char), cudaMemcpyHostToDevice);

cudaMemcpy(d\_sin\_len, &sin\_len, sizeof(int), cudaMemcpyHostToDevice);

cudaMemcpy(d\_sout, sout, (sout\_len + 1) \* sizeof(char), cudaMemcpyHostToDevice);

kernel<<<1, sin\_len>>>(d\_sin, d\_sin\_len, d\_sout);

cudaMemcpy(sout, d\_sout, (sout\_len + 1) \* sizeof(char), cudaMemcpyDeviceToHost);

sout[sout\_len] = '\0';

printf("Sin: %s\n", sin);

printf("Sout: %s\n", sout);

cudaFree(d\_sin);

cudaFree(d\_sin\_len);

cudaFree(d\_sout);

return 0;

}

**Output:**

****