CS 3502
Operating Systems

Project 2 Lab

Kun Suo

Computer Science, Kennesaw State University

https://kevinsuo.github.io/

Project 2

- Read and write parallel program using Pthread
- Learn to use Pthread functions
- User level projects, not kernel code

Assignment 1

- Given two character strings s1 and s2. Write a Pthread program to find out the number of substrings, in string s1, that is exactly the same as s2.

- https://github.com/kevinsuo/CS3502/blob/master/project-pthread.c

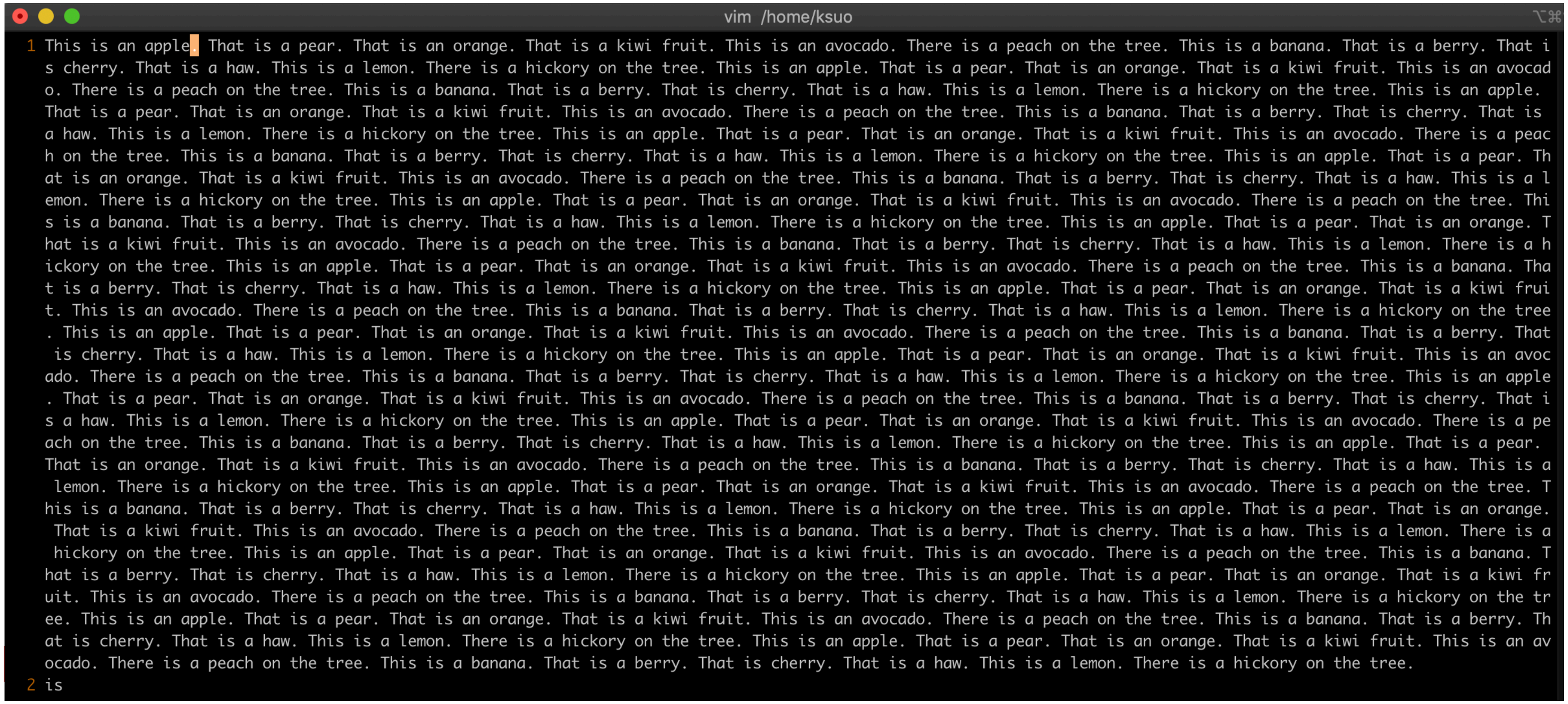
Assignment 1 Examples

- number_substring(“abcdab”, “ab”) = 2,
- number_substring(“aaa”, “a”) = 3,
- number_substring(“abac”, “bc”) = 0.

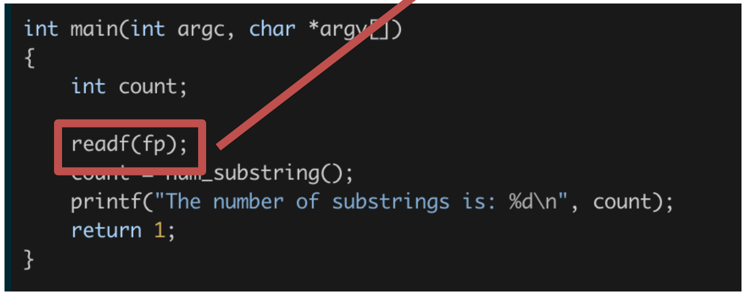
Subsequence not substring

Assignment 1

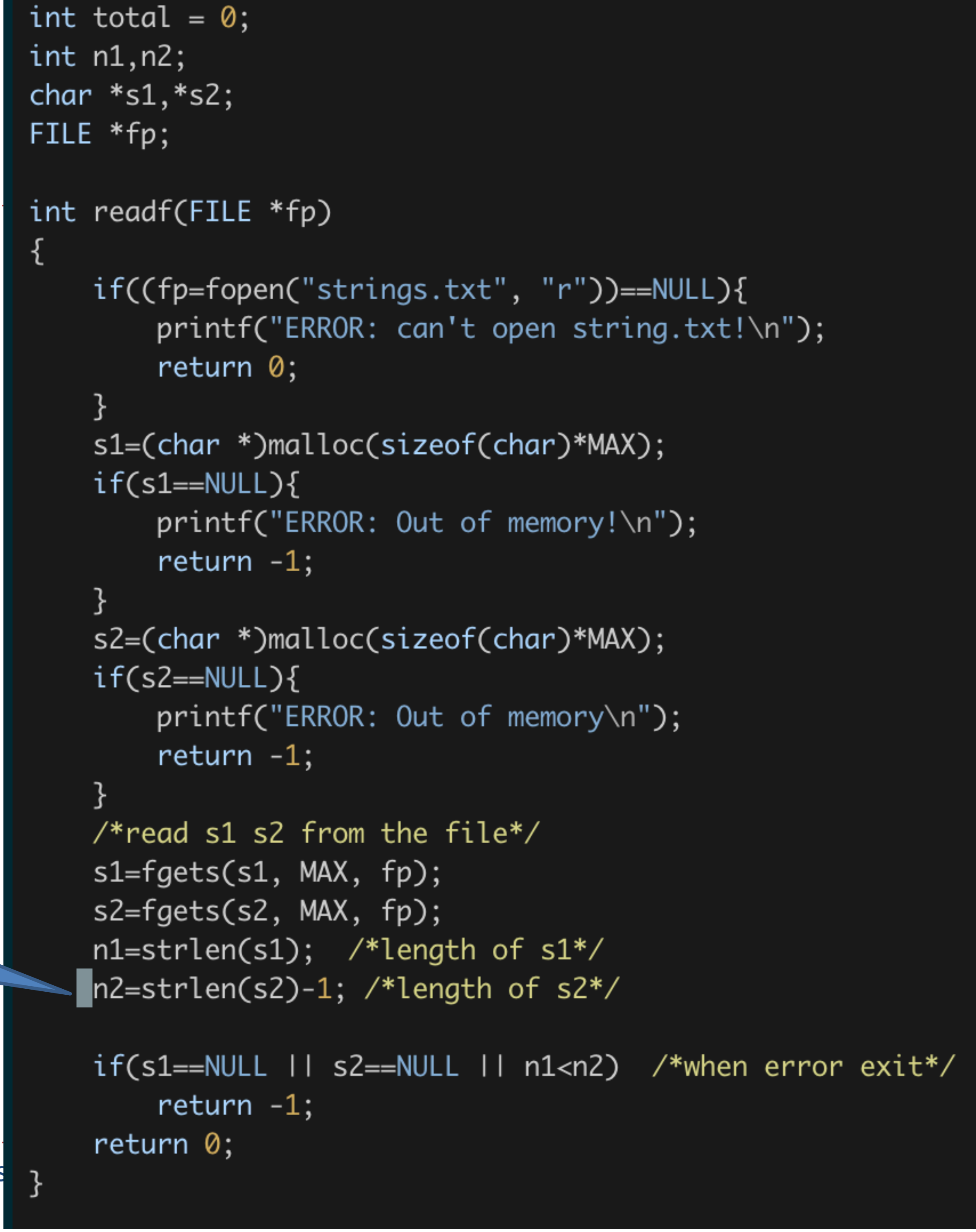
- Input file: https://github.com/kevinsuo/CS3502/blob/master/strings.txt



```
This is an apple. That is a pear. That is an orange. That is a kiwi fruit. This is an avocado. There is a peach on the tree. This is a banana. That is a berry. That i
s cherry. That is a haw. This is a lemon. There is a hickory on the tree. This is an apple. That is a pear. That is an orange. That is a kiwi fruit. This is an avocad
o. There is a peach on the tree. This is a banana. That is a berry. That is cherry. That is a haw. This is a lemon. There is a hickory on the tree. This is an apple.
That is a pear. That is an orange. That is a kiwi fruit. This is an avocado. There is a peach on the tree. This is a banana. That is a berry. That is cherry. That is
a haw. This is a lemon. There is a hickory on the tree. This is an apple. That is a pear. That is an orange. That is a kiwi fruit. This is an avocado. There is a peac
h on the tree. This is a banana. That is a berry. That is cherry. That is a haw. This is a lemon. There is a hickory on the tree. This is an apple. That is a pear. Th
at is an orange. That is a kiwi fruit. This is an avocado. There is a peach on the tree. This is a banana. That is a berry. That is cherry. That is a haw. This is a l
emon. There is a hickory on the tree. This is an apple. That is a pear. That is an orange. That is a kiwi fruit. This is an avocado. There is a peach on the tree. Thi
s is a banana. That is a berry. That is cherry. That is a haw. This is a lemon. There is a hickory on the tree. This is an apple. That is a pear. That is an orange. T
hat is a kiwi fruit. This is an avocado. There is a peach on the tree. This is a banana. That is a berry. That is cherry. That is a haw. This is a lemon. There is a h
ickory on the tree. This is an apple. That is a pear. That is an orange. That is a kiwi fruit. This is an avocado. There is a peach on the tree. This is a banana. Tha
t is a berry. That is cherry. That is a haw. This is a lemon. There is a hickory on the tree. This is an apple. That is a pear. That is an orange. That is a kiwi frui
t. This is an avocado. There is a peach on the tree. This is a banana. That is a berry. That is cherry. That is a haw. This is a lemon. There is a hickory on the tree
. This is an apple. That is a pear. That is an orange. That is a kiwi fruit. This is an avocado. There is a peach on the tree. This is a banana. That is a berry. That
 is cherry. That is a haw. This is a lemon. There is a hickory on the tree. This is an apple. That is a pear. That is an orange. That is a kiwi fruit. This is an avoc
ado. There is a peach on the tree. This is a banana. That is a berry. That is cherry. That is a haw. This is a lemon. There is a hickory on the tree. This is an apple
. That is a pear. That is an orange. That is a kiwi fruit. This is an avocado. There is a peach on the tree. This is a banana. That is a berry. That is cherry. That i
s a haw. This is a lemon. There is a hickory on the tree. This is an apple. That is a pear. That is an orange. That is a kiwi fruit. This is an avocado. There is a pe
ach on the tree. This is a banana. That is a berry. That is cherry. That is a haw. This is a lemon. There is a hickory on the tree. This is an apple. That is a pear.
That is an orange. That is a kiwi fruit. This is an avocado. There is a peach on the tree. This is a banana. That is a berry. That is cherry. That is a haw. This is a
 lemon. There is a hickory on the tree. This is an apple. That is a pear. That is an orange. That is a kiwi fruit. This is an avocado. There is a peach on the tree. T
his is a banana. That is a berry. That is cherry. That is a haw. This is a lemon. There is a hickory on the tree. This is an apple. That is a pear. That is an orange.
 That is a kiwi fruit. This is an avocado. There is a peach on the tree. This is a banana. That is a berry. That is cherry. That is a haw. This is a lemon. There is a
 hickory on the tree. This is an apple. That is a pear. That is an orange. That is a kiwi fruit. This is an avocado. There is a peach on the tree. This is a banana. T
hat is a berry. That is cherry. That is a haw. This is a lemon. There is a hickory on the tree. This is an apple. That is a pear. That is an orange. That is a kiwi fr
uit. This is an avocado. There is a peach on the tree. This is a banana. That is a berry. That is cherry. That is a haw. This is a lemon. There is a hickory on the tr
ee. This is an apple. That is a pear. That is an orange. That is a kiwi fruit. This is an avocado. There is a peach on the tree. This is a banana. That is a berry. Th
at is cherry. That is a haw. This is a lemon. There is a hickory on the tree. This is an apple. That is a pear. That is an orange. That is a kiwi fruit. This is an av
ocado. There is a peach on the tree. This is a banana. That is a berry. That is cherry. That is a haw. This is a lemon. There is a hickory on the tree.
is
```



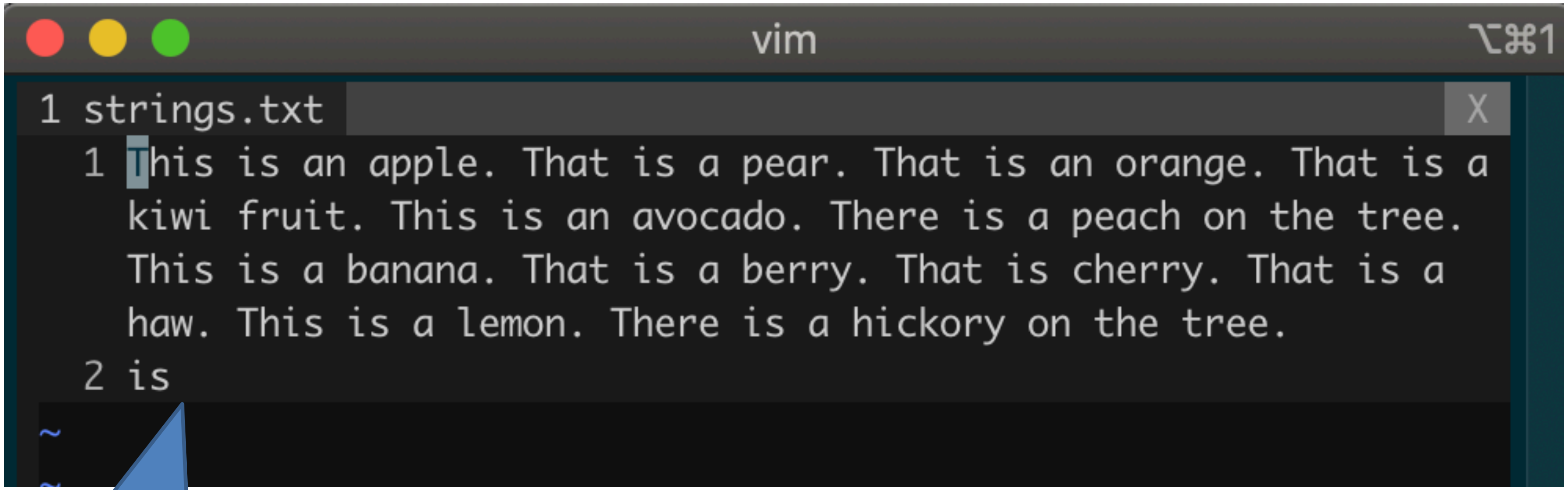
```
int main(int argc, char *argv[])
{
    int count;

    readf(fp);
    count = num_substring();
    printf("The number of substrings is: %d\n", count);
    return 1;
}
```



```
int total = 0;
int n1,n2;
char *s1,*s2;
FILE *fp;

int readf(FILE *fp)
{
    if((fp=fopen("strings.txt", "r"))==NULL){
        printf("ERROR: can't open string.txt!\n");
        return 0;
    }
    s1=(char *)malloc(sizeof(char)*MAX);
    if(s1==NULL){
        printf("ERROR: Out of memory!\n");
        return -1;
    }
    s2=(char *)malloc(sizeof(char)*MAX);
    if(s2==NULL){
        printf("ERROR: Out of memory\n");
        return -1;
    }
    /*read s1 s2 from the file*/
    s1=fgets(s1, MAX, fp);
    s2=fgets(s2, MAX, fp);
    n1=strlen(s1);  /*length of s1*/
    n2=strlen(s2)-1; /*length of s2*/

    if(s1==NULL || s2==NULL || n1<n2)  /*when error exit*/
        return -1;
    return 0;
}
```

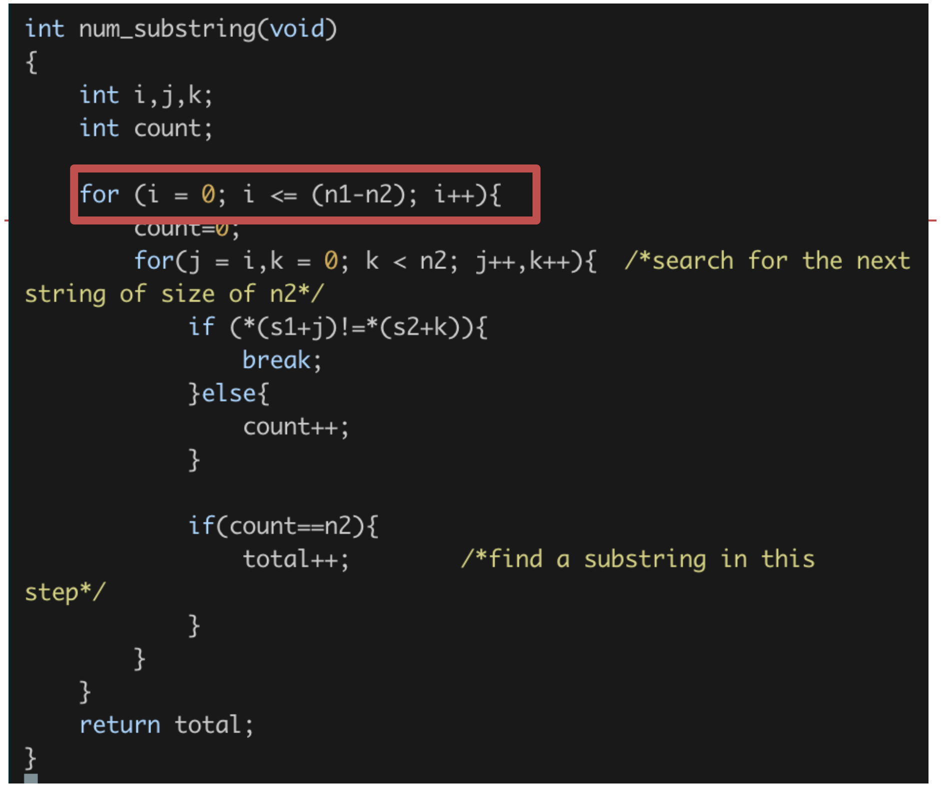
Use strlen(s2)-1
Because there exist a
'\n' at the end of s2

'\n' at the end of s2

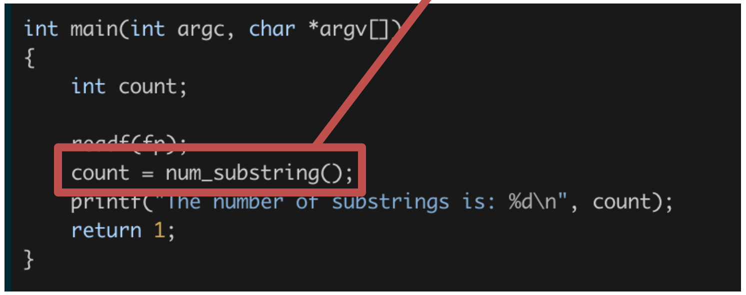


```
1 strings.txt
  1 This is an apple. That is a pear. That is an orange. That is a
    kiwi fruit. This is an avocado. There is a peach on the tree.
    This is a banana. That is a berry. That is cherry. That is a
    haw. This is a lemon. There is a hickory on the tree.
  2 is
~
```

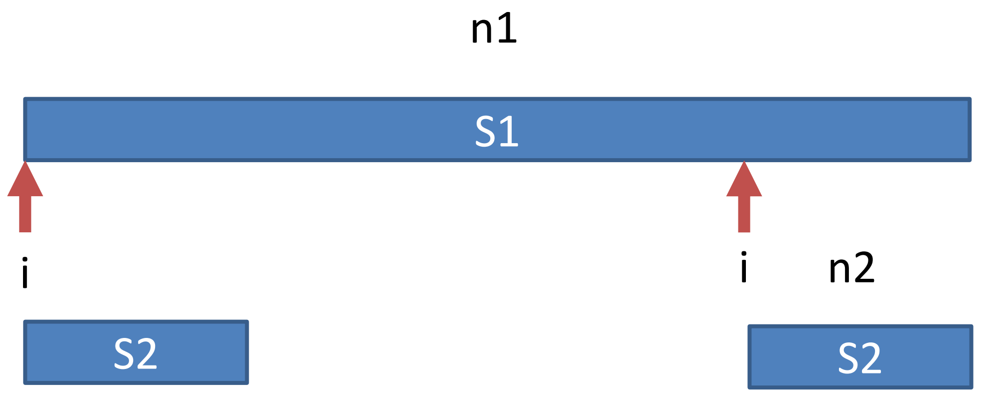
Use strlen(s2)-1
Because there exist a
'EOF' at the end of s2

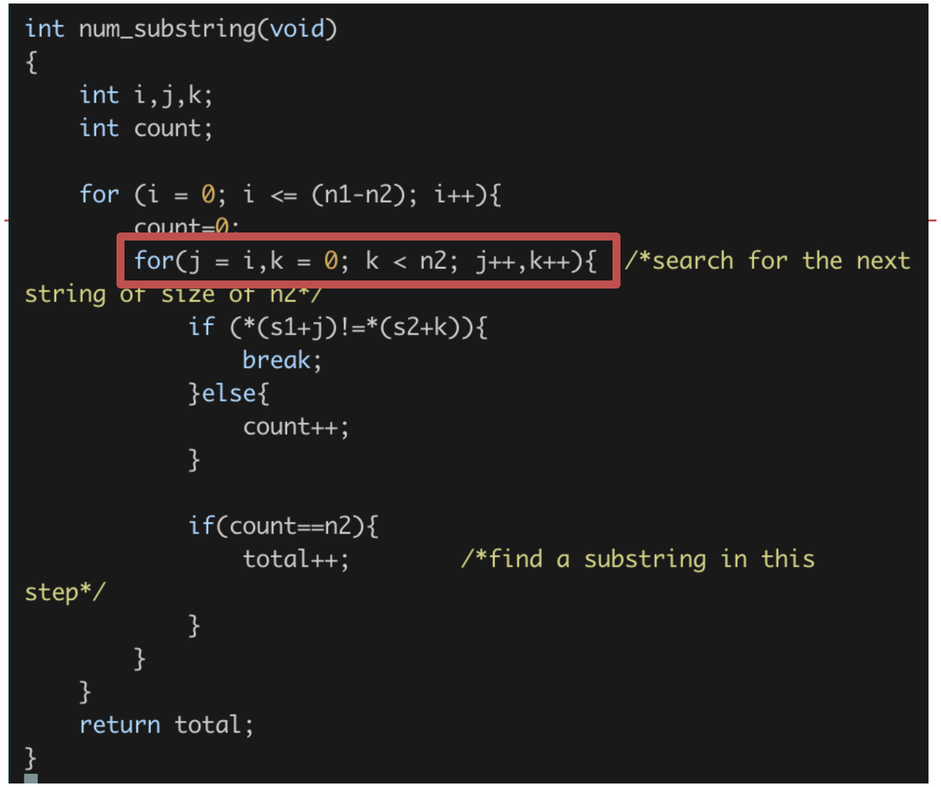


```c
int num_substring(void)
{
    int i,j,k;
    int count;

    for (i = 0; i <= (n1-n2); i++){
        count=0;
        for(j = i,k = 0; k < n2; j++,k++){  /*search for the next
string of size of n2*/
            if (*(s1+j)!=*(s2+k)){
                break;
            }else{
                count++;
            }

            if(count==n2){
                total++;        /*find a substring in this
step*/
            }
        }
    }
    return total;
}
```

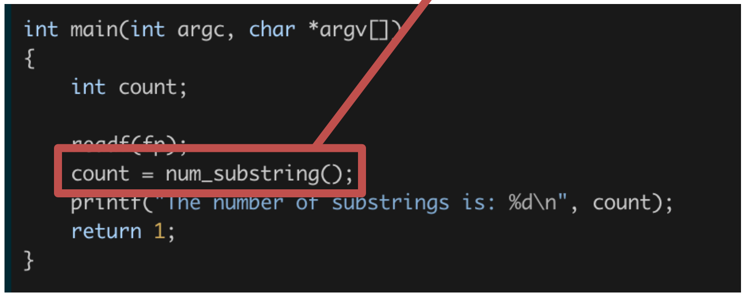


```c
int main(int argc, char *argv[])
{
    int count;

    readf(fp);
    count = num_substring();
    printf("The number of substrings is: %d\n", count);
    return 1;
}
```

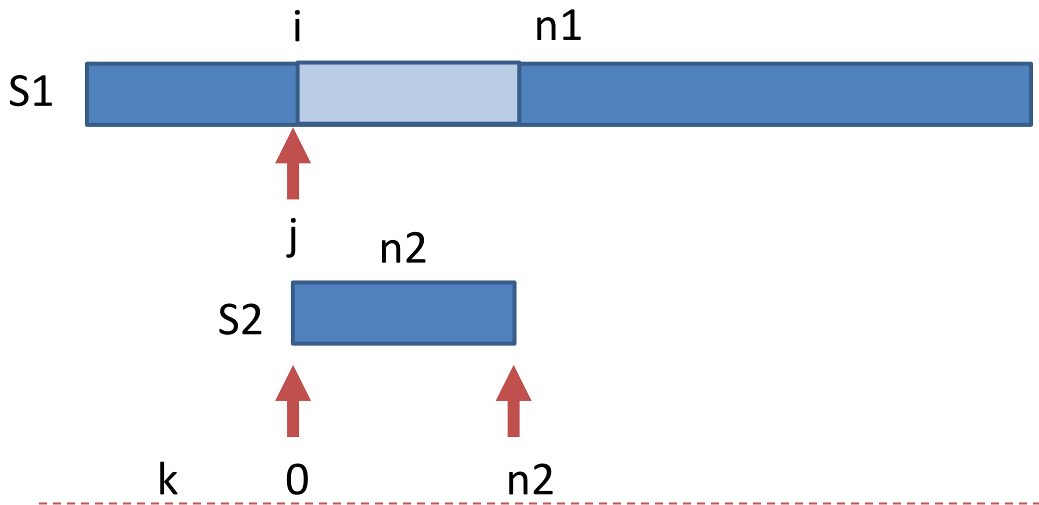


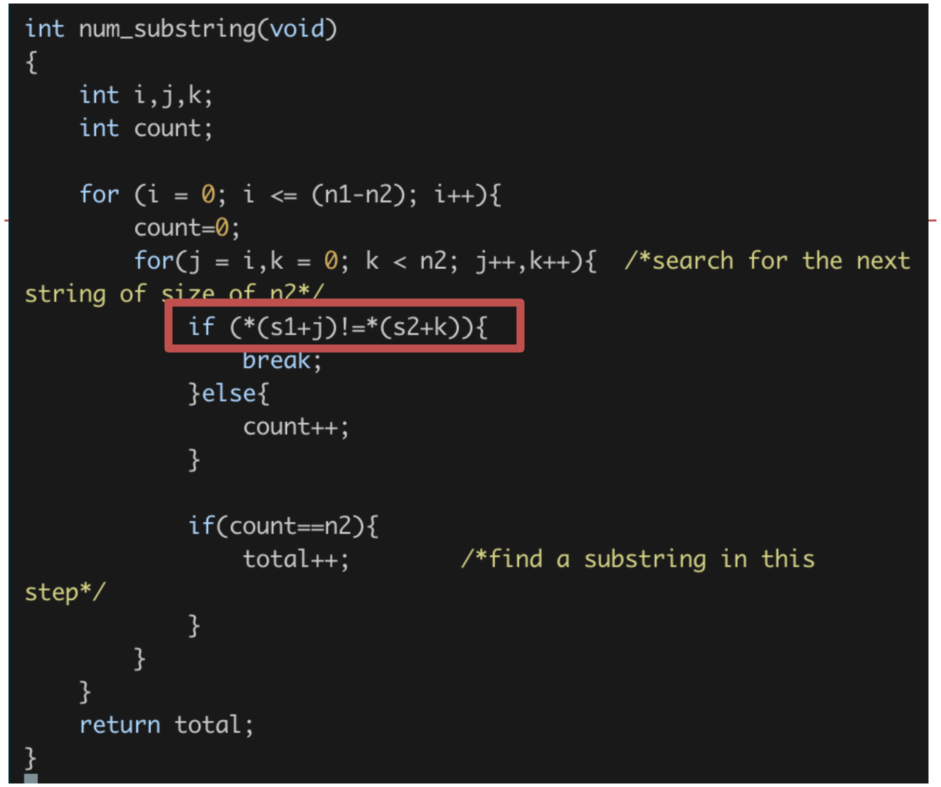


```
int num_substring(void)
{
    int i,j,k;
    int count;

    for (i = 0; i <= (n1-n2); i++){
        count=0;
        for(j = i,k = 0; k < n2; j++,k++){  /*search for the next
string of size of n2*/
            if (*(s1+j)!=*(s2+k)){
                break;
            }else{
                count++;
            }

            if(count==n2){
                total++;        /*find a substring in this
step*/
            }
        }
    }
    return total;
}
```

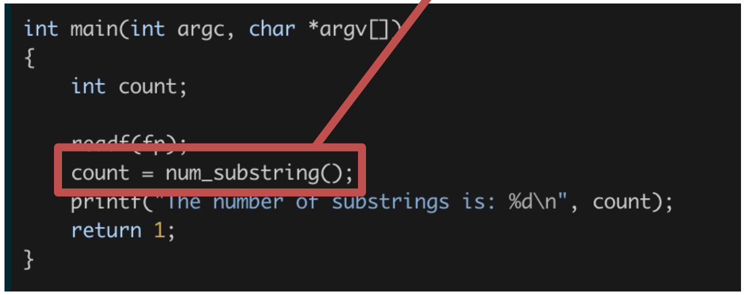


```
int main(int argc, char *argv[])
{
    int count;

    readf(fp);
    count = num_substring();
    printf("The number of substrings is: %d\n", count);
    return 1;
}
```

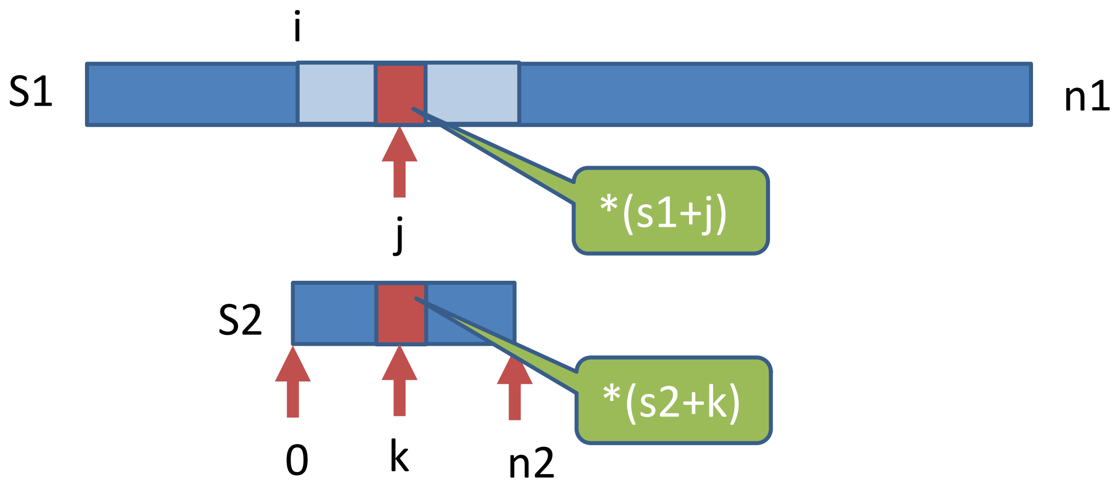


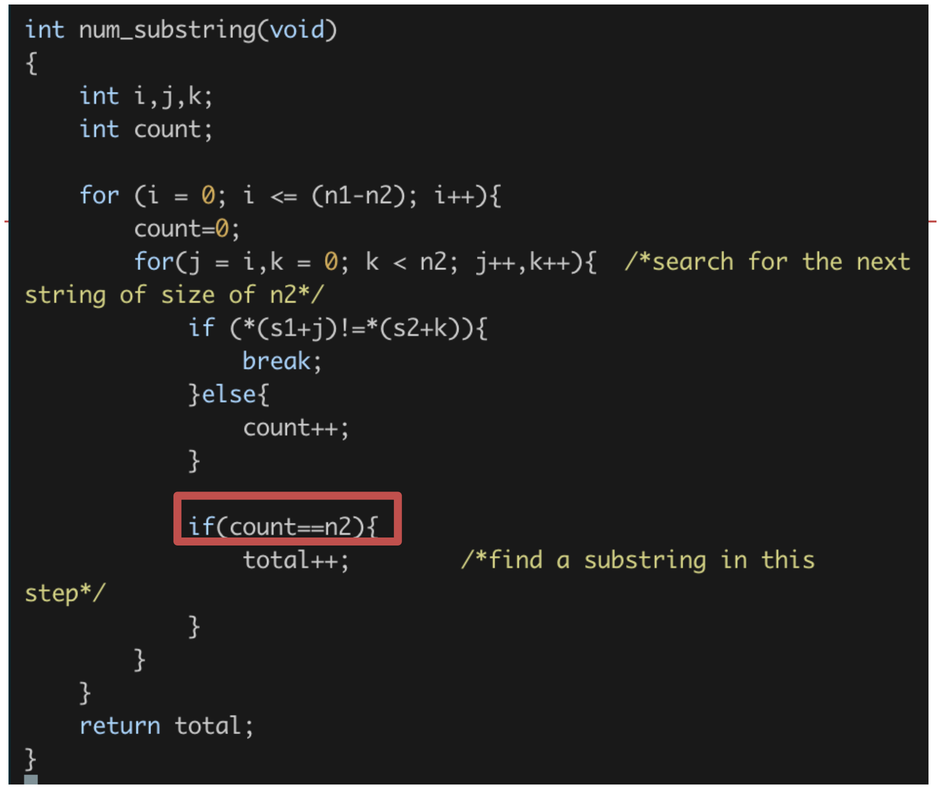


```
int num_substring(void)
{
    int i,j,k;
    int count;

    for (i = 0; i <= (n1-n2); i++){
        count=0;
        for(j = i,k = 0; k < n2; j++,k++){  /*search for the next
string of size of n2*/
            if (*(s1+j)!=*(s2+k)){
                break;
            }else{
                count++;
            }

            if(count==n2){
                total++;        /*find a substring in this
step*/
            }
        }
    }
    return total;
}
```

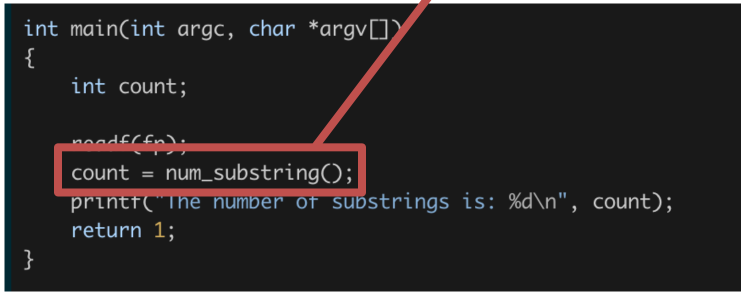


```
int main(int argc, char *argv[])
{
    int count;

    readf(fp);
    count = num_substring();
    printf("The number of substrings is: %d\n", count);
    return 1;
}
```

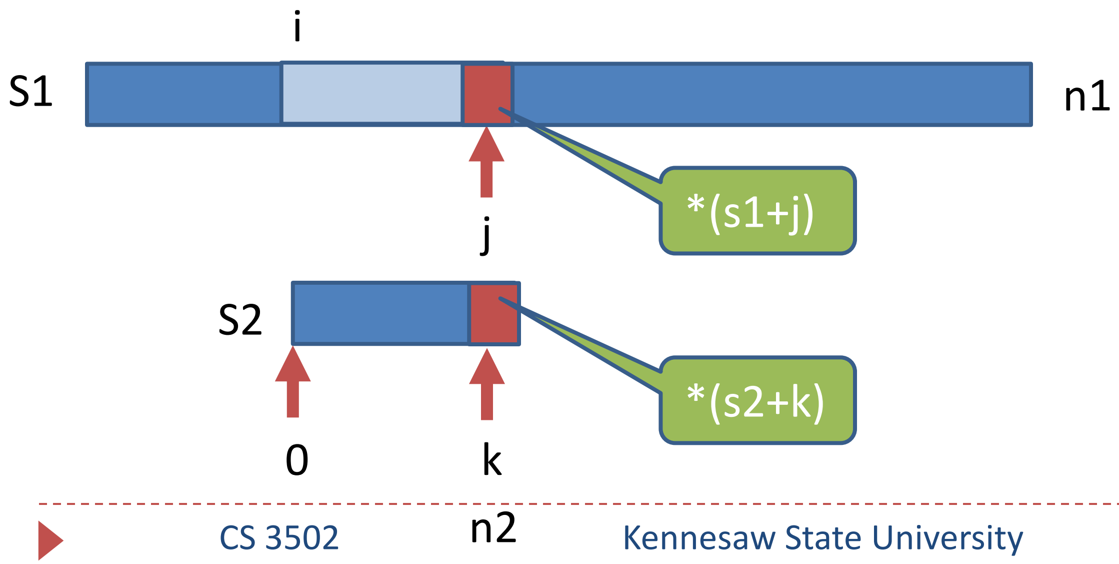




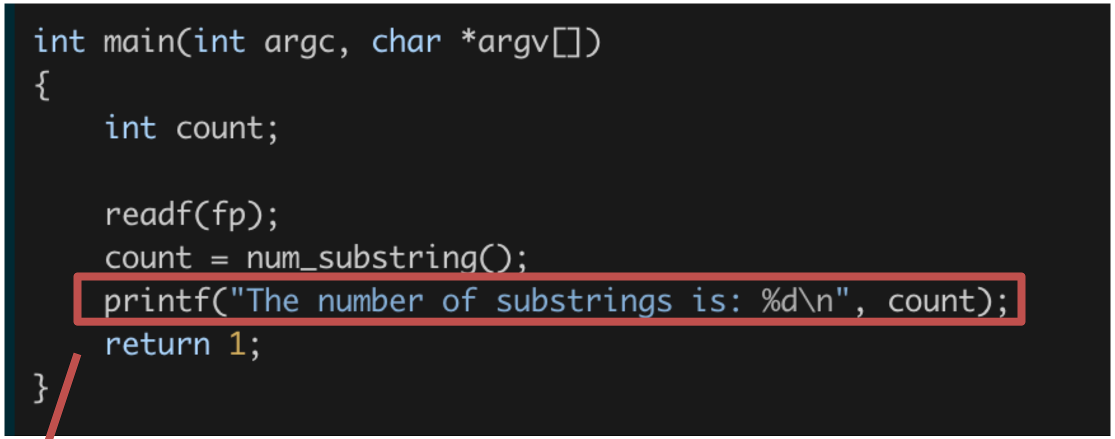
```
int num_substring(void)
{
    int i,j,k;
    int count;

    for (i = 0; i <= (n1-n2); i++){
        count=0;
        for(j = i,k = 0; k < n2; j++,k++){  /*search for the next
string of size n2*/
            if (*(s1+j)!=*(s2+k)){
                break;
            }else{
                count++;
            }

            if(count==n2){
                total++;        /*find a substring in this
step*/
            }
        }
    }
    return total;
}
```



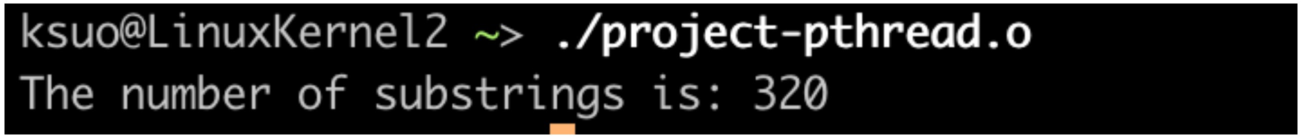
```
int main(int argc, char *argv[])
{
    int count;

    readf(fp);
    count = num_substring();
    printf("The number of substrings is: %d\n", count);
    return 1;
}
```



Assignment 1



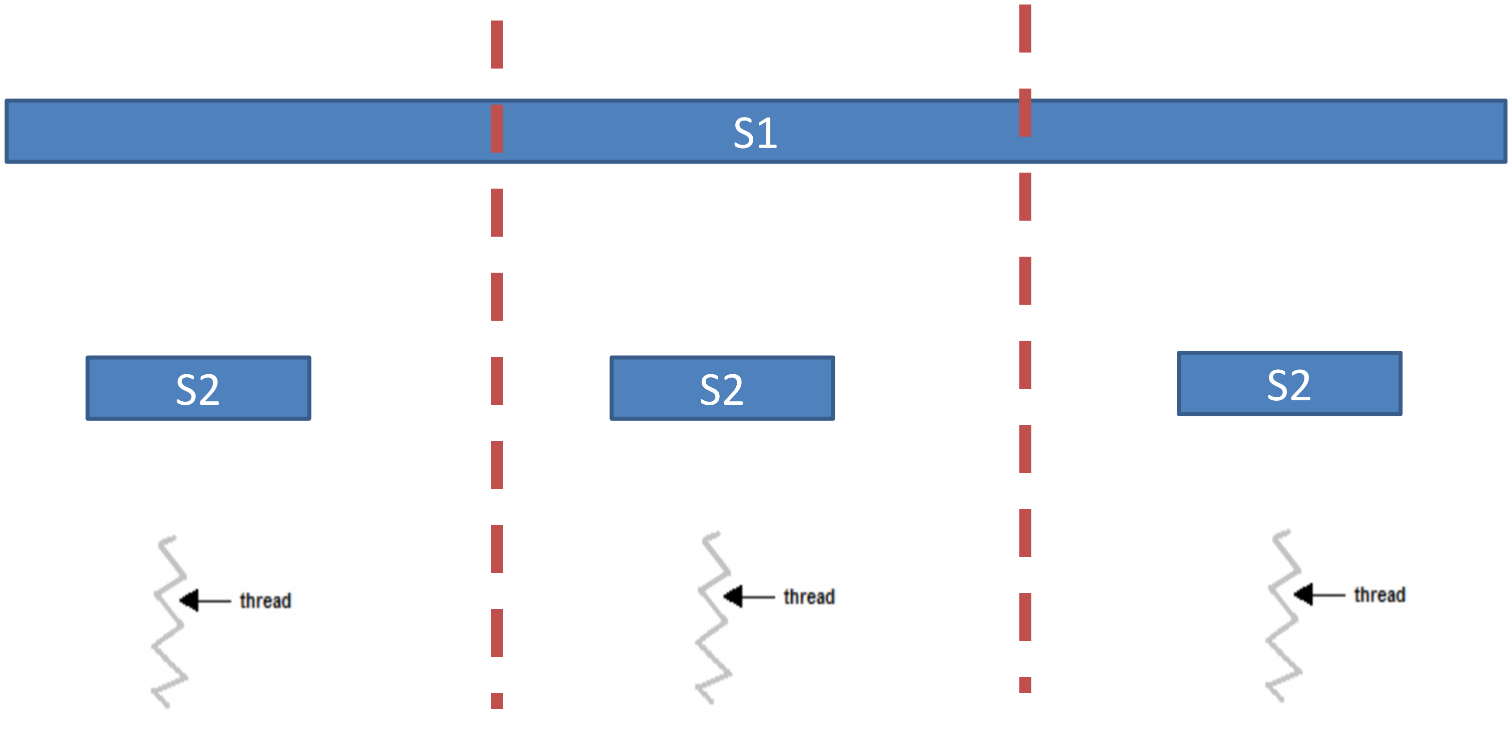
```
int main(int argc, char *argv[])
{
    int count;

    readf(fp);
    count = num_substring();
    printf("The number of substrings is: %d\n", count);
    return 1;
}
```



```
ksuo@LinuxKernel2 ~> ./project-pthread.o
The number of substrings is: 320
```

Assignment 1

- Write a parallel program using Pthread based on this sequential solution.



Assignment 1

pthread_create (thread, attr, start_routine, arg)

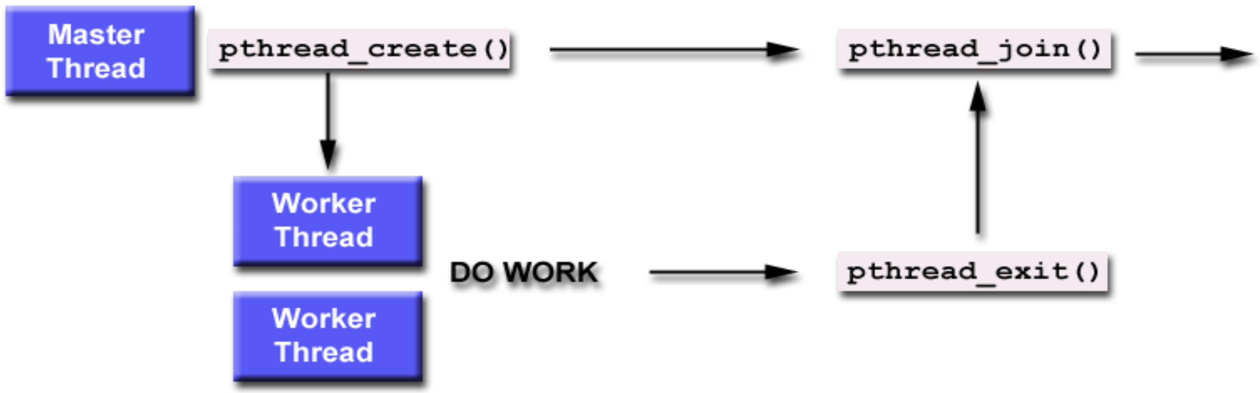
- **creates a thread and makes it executable;**
- **1st parameter: pointer to the thread**
- **2rd parameter: set attributes to threads, usually NULL**
- **3rd parameter: the function for the thread to run**
- **4th parameter: parameter for thread function**

pthread_exit (status)

• **If** main() **finishes before the threads it has created, and exists with the** pthread_exit(), **the other threads will continue to execute. Otherwise, they will be automatically terminated when** main() **finishes**

Assignment 1

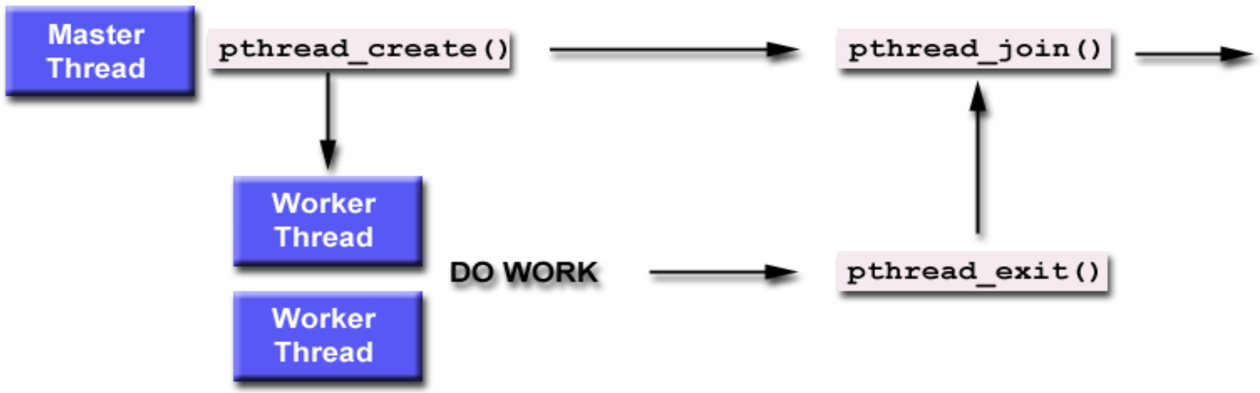
https://github.com/kevinsuo/CS3502/blob/master/parallel-template.c

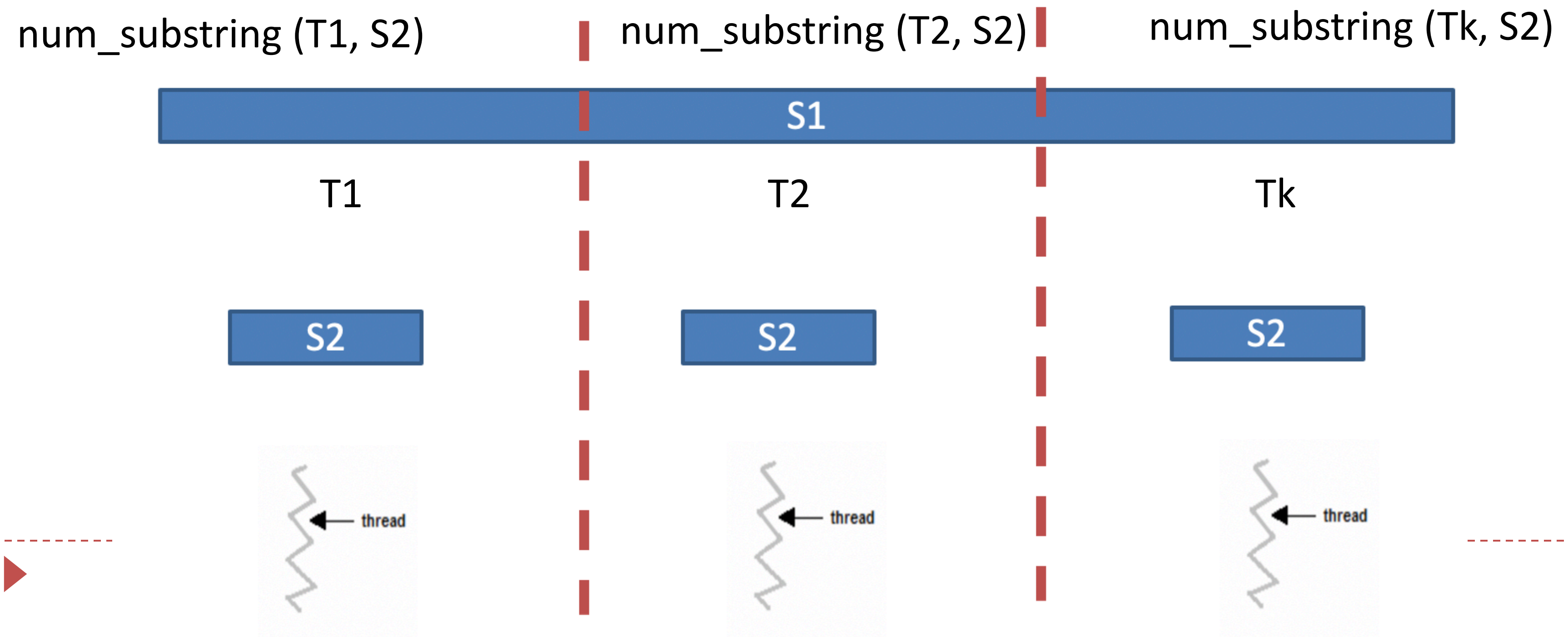


```
Main()
{
        //create multiple threads
        pthread_create

        //wait thread to finish in main
         pthread_join

        //sum up the number from each thread
        sum = num[0]+ num[1]+ … num[k]
}
```

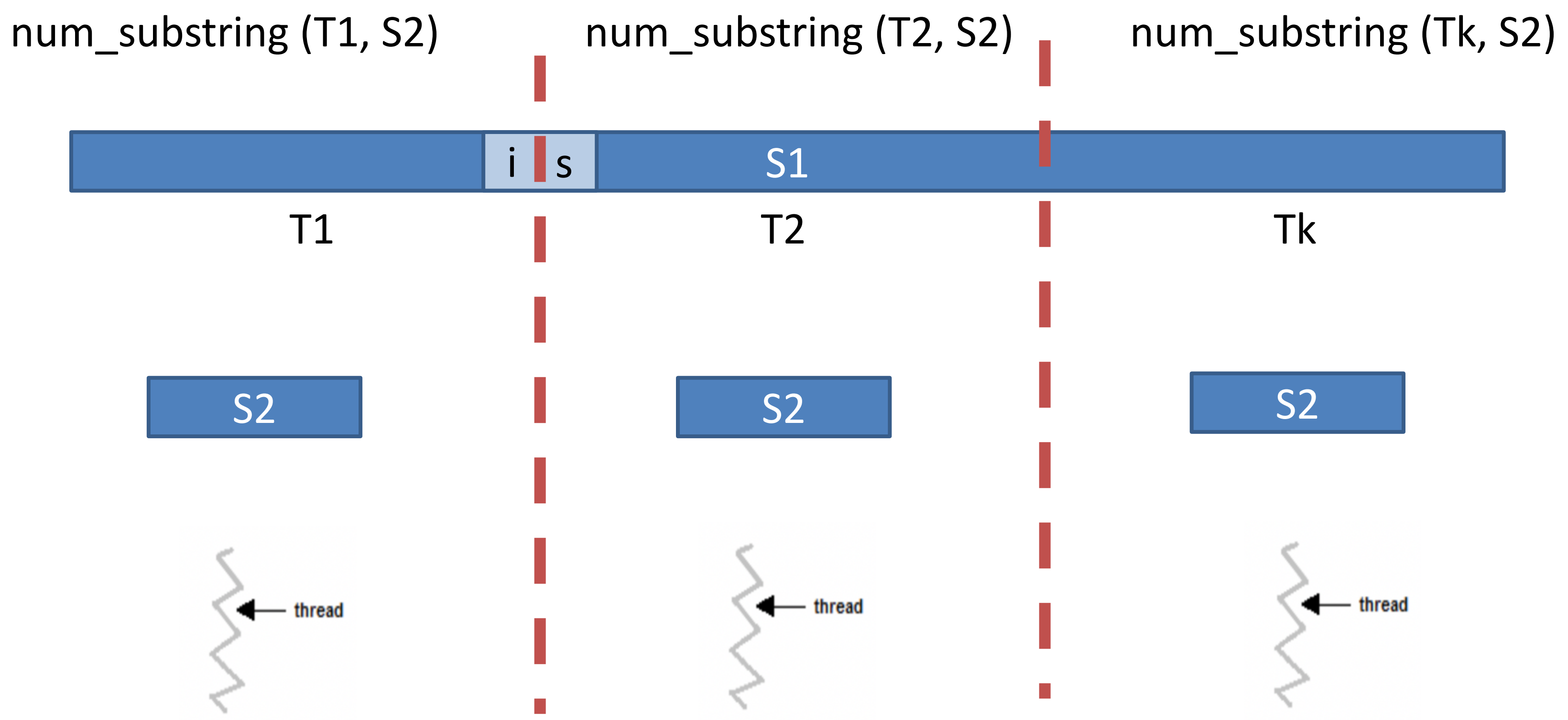
```
thread()
{
        //do something
}
```

Assignment 1



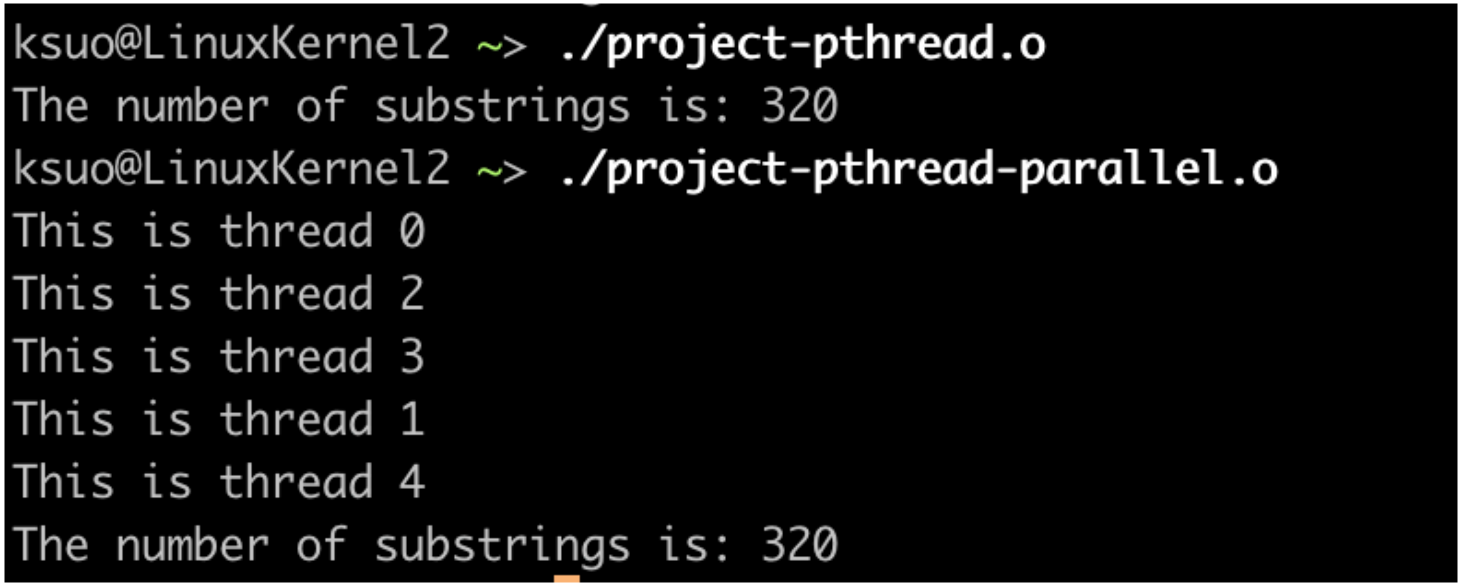


Corner Case in Assignment 1



Verify whether your parallel thread is correct

- Modify the strings.txt by yourself
- Compare the sequential and parallel program results that whether they are the same



```
ksuo@LinuxKernel2 ~> ./project-pthread.o
The number of substrings is: 320
ksuo@LinuxKernel2 ~> ./project-pthread-parallel.o
This is thread 0
This is thread 2
This is thread 3
This is thread 1
This is thread 4
The number of substrings is: 320
```

Submission

1. source code

2. output screenshot of your parallel code

3. a report describe your code logic

Questions

- T/Th 3-4pm, J-318

- Send me emails or make appointments