

CS19001: Programming and Data Structures Laboratory

Soumyajit Dey, Aritra Hazra
CSE, IIT Kharagpur

http://cse.iitkgp.ac.in/~aritrah/course/lab/PDS/Autumn2018/CS19101_PDS-Lab_Autumn2018.html

12-Nov-2018

Programming Assignments

Complete and submit during lab

Assignment 11: Binary Convolution Filter

Write a single C program which will parse command line arguments and decide accordingly to either create a file or process an existing file. The argument specification is as follows.

- 1 Creation of file: `./a.out create n filename.txt`
Your program should open an empty file called "filename.txt" and write a randomly generated binary string of 0s and 1s of size/length n to that file.
- 2 Processing file: `./a.out process filename.txt n m`
Your program should open an existing file called "filename.txt" that contains a binary string of length n , process that string on an element-by-element basis using a *convolution operation* and overwrite the existing string in the same file given as "filename.txt".
The usage of m is described in future slides!

Assignment 11: Binary Convolution Filter

Convolution Operation: For every element b in the binary string, 4 neighbouring elements are as follows:

- two elements immediately preceding b (say, e_1 and e_2)
- two elements immediately succeeding b (say, f_1 and f_2)

Perform a logical OR operation on these 5 values, i.e.

$(e_1|e_2|b|f_1|f_2)$ and overwrite the value of b with the result in the file.

Note that, for elements residing at the boundaries of the string, only consider the neighbours of the element that are there in the string.

Example: Consider the binary string 000111000000. If we apply the operation for every element in the string, the final output is 01111111000.

Note that, for the element at position 1 which is 0 has one preceding neighbour 0 and two succeeding neighbours 0 and 1. For this element, perform OR operation on these 4 elements only.

Assignment 11: Binary Convolution Filter

NOTE: You cannot read the entire string from the file in a character array of size n and perform this operation.

You are allowed to use a character array of size $m < n$ (given as command line argument) for storing a chunk of size m from the file.

Using this array, perform the following operations for every chunk in the file.

- 1 read a chunk of size m from the file
- 2 process m elements using the *convolution operation*
- 3 write back the modified chunk to the same file

Use **only one extra character array** of size m for storing the output of the convolution operation at each step and extra variables for obtaining neighbours of elements that were not a part of the chunk obtained.

Assume that, n is a multiple of m .

Assignment 11: Helper Functions

```
// convert a character to an integer
int ctoi(char c)
{
    int x;
    x = c - '0';
    return x;
}
```

```
// convert an integer to a character
int itoc(int i)
{
    char x;
    x = i + '0';
    return x;
}
```

Illustrative Example

Say, $m = 10, n = 100$. The first 10 elements shall be read to a char array $A = "0000011010"$. We are going to perform the convolution and store the result in a output array B .

- you need to convert each character to an integer
- for the first element, consider only the next two neighbours, you have $B[0] = A[0] | A[1] | A[2] = 0 | 0 | 0 = 0$
- Similarly, $B[1] = A[0] | A[1] | A[2] | A[3] = 0 | 0 | 0 | 0 = 0$ (one preceding and two succeeding),
 $B[2] = A[0] | A[1] | A[2] | A[3] | A[4] = 0 | 0 | 0 | 0 | 0 = 0$,
 $B[3] = A[1] | A[2] | A[3] | A[4] | A[5] = 0 | 0 | 0 | 0 | 1 = 1$
- For the last two elements you need to **separately** read two more elements from the file, e_1, e_2 . $B[8] = A[6] | A[7] | A[8] | A[9] | e_1$, $B[9] = A[7] | A[8] | A[9] | e_1 | e_2$
- Continuing like this, $B = "0001111111"$. We now write back B to the file and then again read the next 10 elements to A and continue

Thank You