

Mahendra Pruitt

CSCI 375

Jinwoo Kim

10/8/2018

Project 1

```
C:\Users\Wick\Documents\Visual Studio 2017\Projects\osproj\Debug>osproj.exe 10 10
Counter input: 10
Counter limit set to 10

Generated random number: 41
Counter is changed by prod: 1

Generated random number: Consuming number: 41
Counter is changed by cons: 2
18467
Counter is changed by prod: 3
Consuming number: 18467
Counter is changed by cons:
Generated random number: 63344

Counter is changed by prod: 5
Consuming number:
Generated random number: 26500
6334
Counter is changed by prod: 6Counter is changed by cons: 7

Consuming number: 26500
Counter is changed by cons: 8

Generated random number: 19169
Counter is changed by prod: 9
Consuming number: 19169
Generated random number: 15724
Counter is changed by cons: 10
Counter is changed by prod: 11
Consumer thread terminating...

Producer thread terminating...
Press any key to continue . . .
```

// osproj.cpp : This file contains the 'main' function. Program execution begins and ends there.

//

```
#include "pch.h"
#include <iostream>
#include <Windows.h>
#include <ctime>
#include <cstdlib>
#include <stdlib.h>
#define MAX_THREADS 2
```

//Mahendra Pruitt CSCI 375 -- OPERATING SYSTEMS PROJECT 1

using namespace std;

```
HANDLE hThreads[MAX_THREADS]; // # of threads
DWORD id[MAX_THREADS]; // array of thread ids
DWORD waiter;
int in = 0, out = 0, buffcount = 0; // used to check how much items are in buffer
int counter = 0; // counter until limit
int cl, buffsize; // counter limit, to be declared later on and changed in the threads
```

```

// later change in program itself to be buffer size
int *buffer;

DWORD WINAPI randProd(LPVOID n) { // producer thread

    while (counter < c1) {

        while (buffcount == buffsize) { ; } // while buffer is full, do nothing

        int r = rand();
        cout << "\nGenerated random number: " << r << endl;
        buffer[in] = r;
        counter++;
        cout << "Counter is changed by prod: " << counter << endl;
        in = (in + 1) % buffsize;
        buffcount++;

    }
    cout << "Producer thread terminating..." << endl;
    return (DWORD)n;
}

DWORD WINAPI randCons(LPVOID n) { // consumer thread
    while (counter < c1) {
        while (buffcount == 0) { ; } // while buffer is empty, do nothing.

        cout << "Consuming number: " << buffer[out] << endl;
        out = (out + 1) % buffsize;
        counter++;
        cout << "Counter is changed by cons: " << counter << endl;
        buffcount--;

    }
    cout << "Consumer thread terminating..." << endl;
    return (DWORD)n;
}

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

    if (argc != 3) {
        // error if 2+ or 2- inputs are entered
        cout << "TWO INPUTS REQUIRED.\NUSAGE:<PROGRAM NAME> <BUFFERSIZE>
<COUNTERLIMIT>\n\nEXITING PROGRAM...\n";
        system("pause");
        return -1;
    }

    srand(time(0)); // initialize randomization
    cout << "Counter input: " << argv[2] << endl;
    c1 = atoi(argv[2]); // set counterlimit to c1
    cout << "Counter limit set to " << c1 << endl;
    buffsize = atoi(argv[1]);
    buffer = new int[buffsize]; // dynamically set the buffer size based on user input

```

```
    hThreads[0] = CreateThread(NULL, 0, randProd, (LPVOID)counter, NULL, &id[0]);  
    //thread 1 for randprod  
    hThreads[1] = CreateThread(NULL, 0, randCons, (LPVOID)counter, NULL, &id[1]);  
    //thread 2 for randcons  
    waiter = WaitForMultipleObjects(MAX_THREADS, hThreads, TRUE, INFINITE); // wait  
    for our threads to stop  
  
    for (int i = 0; i < MAX_THREADS; i++) {  
        CloseHandle(hThreads[i]);  
    }  
  
    system("pause");  
    return 0;  
  
}
```