

Mar 11, 09 11:15

Semephore.cpp

Page 1/2

```

/*****
/
/      filename:  Semephore.cpp
/
/      description:  Implements Semephores for the simulator
/
/      author:      Paladino, Zac
/      login id:    cps346-n1.16
/
/      class:       CPS 346
/      instructor:  Perugini
/      assignment:   PJ #2
/
/      assigned:    February 18, 2009
/      due:         March 11, 2009
/
/*****/

#include <iostream>
#include <iomanip>
#include <queue>
#include <list>
#include <vector>
#include <string>
#include <fstream>
using namespace std;
#include "Functions.h"

struct Process
{
    string Event;
    string RQ;
    int Time, Job, Memory, RT, RTM, RQT, FTime, STime, IOBurst, IOS, IOB;
    bool started, IOclean;
};

struct Semephore
{
    int value;
    list < Process > SemList;
};

void
HandleSem (vector < string > &tokens, Semephore Semephores[],
           list < Process > &CPU, list < Process > &RQ1,
           list < Process > &RQ2, vector < Process > &Finished, int &memory,
           int &count, int &CPURQ1, int &CPURQ2, bool &getcm, int &time,
           ofstream &out, bool &ew, bool &es)
{
    if (count == 1) {
        for (int i = 0; i < 5; i++) {
            Semephores[i].value = 1;
        }
    }
    if (tokens[0] == "W") {
        if (time == StringToInt (tokens[1])) {
            int i = StringToInt (tokens[2]);
            if (Semephores[i].value < 1) {
                Semephores[i].SemList.push_back (CPU.front ());
                CPU.pop_front ();
                CPURQ1 = 100;
                CPURQ2 = 300;
            }
            Semephores[i].value--;
            getcm = true;
            ew = true;
        }
        else {
            getcm = false;
        }
    }
}

```

Mar 11, 09 11:15

Semephore.cpp

Page 2/2

```

    }
}
if (tokens[0] == "S") {
    if (time == StringToInt (tokens[1])) {
        int i = StringToInt (tokens[2]);
        if (!Semephores[i].SemList.empty ()) {
            Semephores[i].SemList.front ().RQ = "RQ1";
            RQ1.push_back (Semephores[i].SemList.front ());
            Semephores[i].SemList.pop_front ();
        }
        Semephores[i].value++;
        getcm = true;
        es = true;
    }
    else {
        getcm = false;
    }
}
}
}

```