CS 354
Lab 1
Manmohit Sehgal

The lab wanted us to implement scheduling policy similar to SOLARIS.

Steps taken :
– Implemented the solaris table
– Modified the file resched. C
– Implemented cpuintensinve.c , iointensive.c, hybridprocess.c

Implementation of the SOLARIS TABLE:

To implement the TS table. I hard coded the values in the file initialize.c according to the time quantum, sleep return and time slice. The TS table provided us with the priority levels that would change accordingly.

CPU-intensive:
Process if CPU-intensive when :
– The PR_STATE is PR_CURR.
– The process has consumed all of its time slice.

CASE 1:
– LOOP 1 < 10; LOOP 2 < 1000

OUTPUT CASE1:

```
 LAB 3 RESULTS!
CPU-Intensive - CPU pid: 2    count: 0    Prio: 20    TS: 120    Clock Counter: 2055999754
CPU-Intensive - CPU pid: 2    count: 1    Prio: 20    TS: 119    Clock Counter: 2055999754
CPU-Intensive - CPU pid: 2    count: 2    Prio: 20    TS: 118    Clock Counter: 2055999754
CPU-Intensive - CPU pid: 2    count: 3    Prio: 20    TS: 117    Clock Counter: 2055999754
CPU-Intensive - CPU pid: 2    count: 4    Prio: 20    TS: 116    Clock Counter: 2055999754
CPU-Intensive - CPU pid: 2    count: 5    Prio: 20    TS: 115    Clock Counter: 2055999754
CPU-Intensive - CPU pid: 2    count: 6    Prio: 20    TS: 114    Clock Counter: 2055999754
CPU-Intensive - CPU pid: 2    count: 7    Prio: 20    TS: 113    Clock Counter: 2055999754
CPU-Intensive - CPU pid: 2    count: 8    Prio: 20    TS: 112    Clock Counter: 2055999754
CPU-Intensive - CPU pid: 2    count: 9    Prio: 20    TS: 111    Clock Counter: 2055999754
CPU-Intensive - CPU pid: 3    count: 0    Prio: 20    TS: 120    Clock Counter: 2071553352
CPU-Intensive - CPU pid: 3    count: 1    Prio: 20    TS: 119    Clock Counter: 2071553352
CPU-Intensive - CPU pid: 3    count: 2    Prio: 20    TS: 118    Clock Counter: 2071553352
CPU-Intensive - CPU pid: 3    count: 3    Prio: 20    TS: 117    Clock Counter: 2071553352
CPU-Intensive - CPU pid: 3    count: 4    Prio: 20    TS: 116    Clock Counter: 2071553352
CPU-Intensive - CPU pid: 3    count: 5    Prio: 20    TS: 115    Clock Counter: 2071553352
CPU-Intensive - CPU pid: 3    count: 6    Prio: 20    TS: 114    Clock Counter: 2071553352
CPU-Intensive - CPU pid: 3    count: 7    Prio: 20    TS: 113    Clock Counter: 2071553352
CPU-Intensive - CPU pid: 3    count: 8    Prio: 20    TS: 112    Clock Counter: 2071553352
CPU-Intensive - CPU pid: 3    count: 9    Prio: 20    TS: 111    Clock Counter: 2071553352
```

```
CPU-Intensive - CPU pid: 4      count: 0      Prio: 20      TS: 120      Clock Counter: 2127104102
CPU-Intensive - CPU pid: 4      count: 1      Prio: 20      TS: 119      Clock Counter: 2127104102
CPU-Intensive - CPU pid: 4      count: 2      Prio: 20      TS: 118      Clock Counter: 2127104102
CPU-Intensive - CPU pid: 4      count: 3      Prio: 20      TS: 117      Clock Counter: 2127104102
CPU-Intensive - CPU pid: 4      count: 4      Prio: 20      TS: 116      Clock Counter: 2127104102
CPU-Intensive - CPU pid: 4      count: 5      Prio: 20      TS: 115      Clock Counter: 2127104102
CPU-Intensive - CPU pid: 4      count: 6      Prio: 20      TS: 114      Clock Counter: 2127104102
CPU-Intensive - CPU pid: 4      count: 7      Prio: 20      TS: 113      Clock Counter: 2127104102
CPU-Intensive - CPU pid: 4      count: 8      Prio: 20      TS: 112      Clock Counter: 2127104102
CPU-Intensive - CPU pid: 4      count: 9      Prio: 20      TS: 111      Clock Counter: 2127104102
CPU-Intensive - CPU pid: 5      count: 0      Prio: 20      TS: 120      Clock Counter: 2182648945
CPU-Intensive - CPU pid: 5      count: 1      Prio: 20      TS: 119      Clock Counter: 2182648945
CPU-Intensive - CPU pid: 5      count: 2      Prio: 20      TS: 118      Clock Counter: 2182648945
CPU-Intensive - CPU pid: 5      count: 3      Prio: 20      TS: 117      Clock Counter: 2182648945
….
….
….
```

Why This happens ?
– The time quantum never expiries.

CASE 2:
– LOOP 1 < 10; LOOP 2 < 10000000;

OUTPUT CASE2:

```
CPU-Intensive - CPU pid: 2   count: 0        Prio: 0        TS: 105      Clock Counter: 2261398426
CPU-Intensive - CPU pid: 3      count: 0      Prio: 0      TS: 105      Clock Counter: 2302852607
CPU-Intensive - CPU pid: 4      count: 0      Prio: 0      TS: 105      Clock Counter: 2440248578
CPU-Intensive - CPU pid: 5      count: 0      Prio: 0      TS: 105      Clock Counter: 2657643593
CPU-Intensive - CPU pid: 2      count: 1      Prio: 0      TS: 129      Clock Counter: 2739039829
CPU-Intensive - CPU pid: 3      count: 1      Prio: 0      TS: 129      Clock Counter: 2780435159
CPU-Intensive - CPU pid: 6      count: 0      Prio: 0      TS: 105      Clock Counter: 2917829398
CPU-Intensive - CPU pid: 4      count: 1      Prio: 0      TS: 129      Clock Counter: 2999224798
CPU-Intensive - CPU pid: 7      count: 0      Prio: 0      TS: 105      Clock Counter: 3160621773
CPU-Intensive - CPU pid: 5      count: 1      Prio: 0      TS: 129      Clock Counter: 3242017424
CPU-Intensive - CPU pid: 2      count: 2      Prio: 0      TS: 153      Clock Counter: 3323412748
CPU-Intensive - CPU pid: 3      count: 2      Prio: 0      TS: 153      Clock Counter: 3364808077
CPU-Intensive - CPU pid: 6      count: 1      Prio: 0      TS: 129      Clock Counter: 3486202316
CPU-Intensive - CPU pid: 4      count: 2      Prio: 0      TS: 153      Clock Counter: 3567596426
CPU-Intensive - CPU pid: 7      count: 1      Prio: 0      TS: 129      Clock Counter: 3728991752
CPU-Intensive - CPU pid: 5      count: 2      Prio: 0      TS: 153      Clock Counter: 3810385611
CPU-Intensive - CPU pid: 2      count: 3      Prio: 0      TS: 177      Clock Counter: 3891780945
CPU-Intensive - CPU pid: 3      count: 3      Prio: 0      TS: 177      Clock Counter: 3933176271
CPU-Intensive - CPU pid: 6      count: 2      Prio: 0      TS: 153      Clock Counter: 4054570503
CPU-Intensive - CPU pid: 4      count: 3      Prio: 0      TS: 177      Clock Counter: 4135964613
CPU-Intensive - CPU pid: 2      count: 4      Prio: 0      TS: 1        Clock Counter: 4177359947
CPU-Intensive - CPU pid: 3      count: 4      Prio: 0      TS: 1        Clock Counter: 4218718868
CPU-Intensive - CPU pid: 7      count: 2      Prio: 0      TS: 153      Clock Counter: 5111505
```

```
CPU-Intensive - CPU pid: 5    count: 3    Prio: 0    TS: 177     Clock Counter: 86454540
CPU-Intensive - CPU pid: 4    count: 4    Prio: 0    TS: 1      Clock Counter: 127815302
CPU-Intensive - CPU pid: 6    count: 3    Prio: 0    TS: 177    Clock Counter: 329157064
CPU-Intensive - CPU pid: 5    count: 4    Prio: 0    TS: 1      Clock Counter: 370533637
CPU-Intensive - CPU pid: 2    count: 5    Prio: 0    TS: 25     Clock Counter: 451875408
CPU-Intensive - CPU pid: 3    count: 5    Prio: 0    TS: 25     Clock Counter: 493236168
CPU-Intensive - CPU pid: 7    count: 3    Prio: 0    TS: 177    Clock Counter: 574595851
CPU-Intensive - CPU pid: 6    count: 4    Prio: 0    TS: 1      Clock Counter: 615973583
CPU-Intensive - CPU pid: 4    count: 5    Prio: 0    TS: 25     Clock Counter: 697315347
CPU-Intensive - CPU pid: 7    count: 4    Prio: 0    TS: 1      Clock Counter: 858676118
CPU-Intensive - CPU pid: 5    count: 5    Prio: 0    TS: 25     Clock Counter: 940018118
CPU-Intensive - CPU pid: 2    count: 6    Prio: 0    TS: 49     Clock Counter: 1021378889
CPU-Intensive - CPU pid: 3    count: 6    Prio: 0    TS: 49     Clock Counter: 1062756936
CPU-Intensive - CPU pid: 6    count: 5    Prio: 0    TS: 25     Clock Counter: 1184133908
CPU-Intensive - CPU pid: 4    count: 6    Prio: 0    TS: 49     Clock Counter: 1265510735
….
….
….
….
```

We can clearly see that the output is different because the quantum expiries which switches the processes.

I/O Intensive :

– The process is I/O intensive when the PR_STATE is PR_SLEEP
– A process gives us CPU voluntary.

CASE 1:
LOOP 1 < 10; sleepms(5)
LAB 3 RESULTS!

```
I/O Intensive - CPU pid: 2 + count: 0 + Prio: 52 + TS: 40 Clock Counter: 2052927669
I/O Intensive - CPU pid: 3 + count: 0 + Prio: 52 + TS: 40 Clock Counter: 2054398861
I/O Intensive - CPU pid: 4 + count: 0 + Prio: 52 + TS: 40 Clock Counter: 2055884978
I/O Intensive - CPU pid: 5 + count: 0 + Prio: 52 + TS: 40 Clock Counter: 2057371076
I/O Intensive - CPU pid: 6 + count: 0 + Prio: 52 + TS: 40 Clock Counter: 2058857174
I/O Intensive - CPU pid: 7 + count: 0 + Prio: 52 + TS: 40 Clock Counter: 2060343274
I/O Intensive - CPU pid: 2 + count: 1 + Prio: 58 + TS: 40 Clock Counter: 2061829433
I/O Intensive - CPU pid: 3 + count: 1 + Prio: 58 + TS: 40 Clock Counter: 2063315502
I/O Intensive - CPU pid: 4 + count: 1 + Prio: 58 + TS: 40 Clock Counter: 2064801597
I/O Intensive - CPU pid: 5 + count: 1 + Prio: 58 + TS: 40 Clock Counter: 2066287688
I/O Intensive - CPU pid: 6 + count: 1 + Prio: 58 + TS: 40 Clock Counter: 2067773777
I/O Intensive - CPU pid: 6 + count: 2 + Prio: 58 + TS: 40 Clock Counter: 2070261131
I/O Intensive - CPU pid: 2 + count: 2 + Prio: 58 + TS: 40 Clock Counter: 2071730698
```

For I/O intensive the priority increases always.

Starvation:
– Process starts if the one process takes all the cpu time

In case of hybridprocess, it takes a lot of cpu cycles by bumping up the prioroty of the its processes. Beings a hybrid process as soon it is about to reach its end, it turn into and I/O process to bump up the prioroty. Once that happens the process keeps running starving the CPU.

To fix this problem I decrees the priority of the hybrid process by the following

```
if( preempt < tstab[ptold->prprio].ts_quantum-15){
        ptold->prprio = tstab[ptold->prprio].ts_tqexp;
}
else{
        ptold -> prprio = tstab[ptold->prprio].ts_slpret;
}
```

If the preempt is less than the initial time slice - 15, I make it context switch to cpu so that it does not hog the CPU.

The above code fixes the problem of starvation when a lot of processes are scheduled. Where the hybird process and cpu intensive process run in a way that each other doesnot hog the cpu.