

OS HW 3

Y.15 (a) 8

(b) 6

5.20 (a) number-of-processes

(b) acquire() needs to be called upon entering a function.
release() should be called before leaving a function.

(c) No.

5.32 code in file 5.32.c

5.35

monitor alarm &

condition c;

void delay(int ticks){

int begin_time = read_clock();

while (read_clock() < begin_time + ticks)

cwait();

}

void tick(){

c.signal();

}

}

arrival time - time

6.16

(a) FCFS: 0 2 3 11 15 20

SJF: 0 3 7 12 20

Nonpreemptive SJF: 0 2 3 11 15 20

RR (Quantum=2): 0 1 3 7 12 20

	(b) Turnaround	(c) Waiting
P1	2 - 2 = 0	0 - 2 = 2
P2	3 - 1 = 2	2 - 1 = 1
P3	11 - 4 = 7	7 - 8 = 1
P4	15 - 2 = 13	15 - 4 = 9
P5	20 - 3 = 17	17 - 5 = 12

(d) FCFS, SJF, and RR have the minimum waiting time.

7.8 (a) A deadlock cannot occur with preemption

(b) Yes. A process can't acquire all the resources needed if they're preemptive by a series of requests.