

**Prova scritta 23/07/2019**

Matricola	1	2	3	4	Tot
	/9	/6	/9	/6	/30

Il candidato risolva al meglio gli esercizi proposti, nel tempo massimo di 3 ore. Non è consentito avvalersi di nessuno strumento elettronico, né di libri, appunti o altro sussidio.

**1. 9 punti (obbligatorio)**

un'area di servizio dispone di N distributori di benzina,  $M < N$  addetti al rifornimento di carburante e  $P < M$  addetti al cambio d'olio. Una nuova macchina arrivata nell'area di servizio, si inserisce in una delle code corrispondenti ai distributori. Quando arriva il suo turno, comunica all'addetto al rifornimento la quantità di carburante e se intende cambiare l'olio. Terminato il rifornimento, se richiesto, l'addetto al rifornimento comunica ad un addetto al cambio d'olio di effettuare il servizio. In ogni caso, finito il rifornimento, l'automobile libera il distributore.

Si fornisca una soluzione usando semafori e processi e discutere la soluzione proposta in termini di possibili situazioni di starvation/deadlock.

**2. 6 punti**

Determinare gli intervalli dei valori interi di X, Y e Z per i quali il sistema si trova in uno stato sicuro:

Processo	Risorse allocate				Risorse massime				Risorse disponibili			
	A	B	C	D	A	B	C	D	A	B	C	D
P0	4	Y-1	3	2	6	4	5	5	2	3	10	8
P1	8	0	Z-2	2	10	1	6	8				
P2	4	0	0	0	6	2	0	4				
P3	0	1	2	2	2	3	6	2				
P4	2	1	X+1	3	3	2	4	5				

Determinare gli intervalli dei valori interi di X, Y e Z per i quali il sistema si trova in uno stato sicuro, e se la richiesta di P0 (1, 0, 3, 2) può essere soddisfatta.

**3. 9 punti (obbligatorio)**

Scrivere un programma C che crea M thread produttori e N thread consumatori (con  $M=2*N$ ), con M ed N presi come parametro da riga di comando. Tutti i thread condividono una variabile intera. Ogni thread produttore incrementa di due unità la variabile condivisa se e solo se il valore della stessa è inferiore a 100. Ogni thread consumatore decrementa di due unità la variabile condivisa se e solo se il valore della stessa è superiore a 25. Tutti i thread, dopo l'operazione sulla variabile condivisa (di incremento o di decremento) dormono per 2 secondi. Per implementare la sincronizzazione utilizzare i semafori Posix basati su nome.

**4. 6 punti**

Si realizzi un programma in C e Posix sotto Linux che preso come parametro da riga di comando una directory, analizzi tale directory e ne stampi il nome dei link simbolici e dei corrispondenti file puntati e per cui questi ultimi ne stampi anche il numero di hard link