Luciano Da Silva Moreira

(punti totali: 52.5/87)

+1/1/60+

Sistemi Operativi PAP
2018
Prova Scritta

Nome e Cognome: Luciano Moreira

Tempo a disposizione: 90 minuti / Documentazione ammessa: 1 foglio A5 manoscritto.

Domande a risposta multipla [60 punti]

Ogni domanda può avere una o più risposte corrette. Una risposta completamente corretta è valutata 3 punti.

Domanda 1 ◇ L'algoritmo di scheduling *Multilevel Feedback Queue (MLFQ)* risolve un problema di...

0/3

- [x] *short term scheduling*
- [ ] *medium term scheduling*
- [ ] *long term scheduling*
- [x] *realtime scheduling*

Domanda 2 ◇ In un sistema che implementa un'architettura di tipo NUMA...

3/3

- [x] ogni CPU dispone di una parte privata di memoria
- [x] i tempi di accesso alla memoria possono variare a dipendenza della CPU che effettua la richiesta
- [ ] i tempi di accesso alla memoria sono costanti
- [ ] i tempi di accesso alla memoria non dipendono dalla presenza di cache

Domanda 3 ◇ In un programma multi-thread...

0/3

- [x] ogni thread dispone del proprio stack
- [x] lo stato di esecuzione può essere diverso per ogni thread
- [x] ogni thread dispone del proprio spazio di indirizzamento
- [ ] ogni thread dispone del proprio heap

Domanda 4 ◇ Le funzioni *async safe*...

0/3

- [x] sono eseguite in modo atomico
- [x] possono essere richiamate da thread diversi senza rischio di deadlock
- [x] sono eseguite in modo asincrono
- [x] possono essere utilizzate nei *signal handlers*

Domanda 5 ◇ In un sistema che implementa lo scheduling Fair Share ho 3 utenti U1, U2 e U3 e un programma per il calcolo della meteo P con tempo di esecuzione costante T. Contemporaneamente, l'utente U3 mette in esecuzione un'istanza del programma P, l'utente U2 mette in esecuzione 4 istanze del programma P, mentre U1 mette in esecuzione 2 istanze del programma P. In una situazione del genere...

- [x] al termine dell'esecuzione ogni istanza di P avrà ricevuto complessivamente lo stesso tempo CPU
- [ ] tutte le istanze del programma P di tutti gli utenti termineranno allo stesso momento

1/3

- [x] al termine dell'esecuzione di tutte le istanze di P il tempo CPU ricevuto da ogni utente sarà diverso
- [x] le istanze dell'utente U1 termineranno dopo l'istanza dell'utente U3

Domanda 6 ◇ L'esecuzione pseudo-parallela dei processi è possibile...

- [x] su sistemi con multi-tasking preemptive
- [ ] solo su sistemi con CPU multicore

3/3

- [x] su sistemi con multi-tasking cooperativo
- [ ] solo su sistemi con CPU a un solo core

Domanda 7 ◇ I segnali POSIX...

- [x] possono essere gestiti in modo sincrono
- [ ] sono sincroni

3/3

- [ ] vengono utilizzati per le chiamate di sistema
- [x] sono asincroni

Domanda 8 ◇ La maschera dei segnali (*signal mask*)...

- [ ] permette di associare uno o più segnali ad un gestore (*handler*)
- [ ] gestisce i segnali nello stato *delivered*

3/3

- [x] può essere configurata dal processo/programma
- [ ] permette di bloccare ogni tipo di segnale in entrata (sia normale che realtime)

Domanda 9 ◇ Il segnale SIGINT (i.e. CTRL+C)...

- [x] può essere associato ad un gestore (handler)
- [x] può essere bloccato modificando la maschera

1/3

- [ ] può essere inviato solamente a un processo eseguito in modalità privilegiata
- [x] ha priorità più alta rispetto ad un segnale realtime

Domanda 10 ◇ In un sistema che implementa lo scheduling a lotteria abbiamo 4 processi A,B,C e D con le seguenti priorità: A (priorità 4), B (priorità 3), C (priorità 2), D (priorità 1). La priorità minima è 1, la priorità massima è 5. Lo scheduler è pre-emptive e assegna un quanto di 25ms. Se lasciamo in esecuzione questi processi per un tempo totale complessivo di 50 minuti, con buona probabilità...

1.5/3

- [x] B avrà ottenuto la CPU per più di 10 minuti
- [ ] C avrà ottenuto la CPU per circa di 20 minuti
- [x] D avrà ottenuto la CPU per circa 10 minuti
- [x] A avrà ottenuto la CPU per più di 15 minuti

Domanda 11 ◇ In un sistema realtime di tipo *event driven*...

1/3

- [x] posso utilizzare l'algoritmo di scheduling EDF
- [ ] posso utilizzare l'algoritmo di scheduling RMA
- [x] non posso determinare il laxity time
- [x] i task sono di tipo aperiodico

Domanda 12 ◇ La latenza nei sistemi realtime..

3/3

- [ ] influenza solo i task aperiodici
- [ ] può essere eliminata scegliendo un algoritmo di scheduling appropriato
- [ ] impedisce la schedulabilità dei task
- [x] deve essere presa in considerazione per poter garantire il rispetto delle deadline

Domanda 13 ◇ Per implementare il multi-tasking preemptive è necessario disporre di un meccanismo che...

0/3

- [ ] garantisca che il kernel venga eseguito su un core diverso rispetto ai processi utente
- [x] garantisca che il kernel possa ottenere periodicamente il controllo della CPU
- [ ] permetta ai processi di rilasciare volontariamente la CPU (yield)
- [ ] garantisca che il kernel venga eseguito con una priorità più alta rispetto ai processi utente

Domanda 14 ◇ Subito dopo un **fork**, (quindi prima di un eventuale **exec**)...

2/3

- [x] il PID del processo figlio è sempre numericamente più grande rispetto a quello del padre
- [ ] il processo padre e il processo figlio condividono lo stesso stack
- [x] il contenuto dello spazio di indirizzamento del processo padre è identico a quello del figlio
- [ ] il PID del processo figlio è sempre numericamente più piccolo rispetto a quello del padre

Domanda 15 ◇ Un sistema operativo basato su microkernel...

3/3

- [x] esegue solo una parte minimale delle sue funzionalità in modalità privilegiata
- [ ] esegue tutte sue le funzionalità in modalità privilegiata
- [ ] permette l'accesso diretto all'hardware a partire dalla modalità utente
- [x] si basa sul principio del privilegio minimo

Domanda 16 ♢ In un sistema realtime è più probabile ottenere la *schedulabilità* con...

1/3

- [x] dei valori di *slack time* grandi
- [x] utilizzando un algoritmo con priorità dinamiche
- [x] dei valori di *laxity time* grandi
- [ ] utilizzando un algoritmo con priorità statiche

Domanda 17 ♢ Lo stato di un processo (i.e. *ready*, *running* e *blocked*)...

1.5/3

- [ ] può essere modificato da ogni thread del processo
- [x] viene utilizzato dall'algoritmo di scheduling
- [x] può essere modificato solo dal kernel
- [ ] può essere modificato da solo dal thread principale del processo

Domanda 18 ♢ In un sistema multiprocessore l'*hard affinity*...

3/3

- [ ] permette di ottenere un migliore utilizzo delle risorse di calcolo
- [ ] può essere implementata solo attraverso un algoritmo di scheduling a coda multipla
- [ ] permette ai thread di migrare tra CPU/Core diversi
- [x] introduce dei vincoli all'algoritmo di scheduling

Domanda 19 ♢ L'algoritmo Shortest Remaining Time Next (SRTN)...

1.5/3

- [ ] equivale a FIFO se l'esecuzione dei processi non viene sospesa o prelazionata (pre-emption)
- [x] può portare a una situazione di starvation
- [x] minimizza il tempo di risposta
- [x] necessita di una stima del tempo totale di esecuzione dei processi

Domanda 20 ♢ Le chiamate di sistema...

0/3

- [x] sono necessarie per l'esecuzione di qualsiasi programma
- [x] definiscono l'API del sistema operativo
- [x] sfruttano gli interrupt asincroni
- [ ] sfruttano gli interrupt del *timer*

Domande a risposta aperta [9 punti]

Rispondere ad ogni domanda in modo preciso e conciso

Domanda 21 Qual'è la differenza tra *multitasking cooperativo* e *multitaskint pre-emptive* **[3 punti]**

☐w ☐p- ☒p+ ☐c 2/3

Nel cooperativo si aspetta che il processo/thread rilasce volontariamente (Yield) suo accesso a CPU dopo aver finito suo job mentre pre emptive è forzado a farlo dopo ogni "quantum" di tempo stabilito. (come? da chi?

Domanda 22 Perché i processi di tipo *IO-Bound* possono portare un sistema ad avere più chiamate allo scheduler rispetto a processi di tipo *CPU-Bound*? **[3 punti]**

☐w ☒p- ☐p+ ☐c 1/3

Perché loro iterazione con il S.O. dipende dalle I/O ricevute, (✓) e quindi per ottimizzare la loro esecuzione si fa necessaria l'intervento dello scheduler più frequentemente.

Domanda 23 Uno degli obiettivi generali di un algoritmo di scheduling è l'equità (*fairness*). Puoi fare un esempio di algoritmo/situazione che non rispetta questo obiettivo? **[3 punti]**

☒w ☐p- ☐p+ ☐c

0/3

Nel caso del Round Robin senza priorità si può portare ad una situazione di Starvation dei processi con la priorità più bassa ??

24→ $\left(\frac{1}{7} + \frac{1}{4} = 0,39 \leq 0,77 \text{ quindi, RMA}\right)$ $\frac{1}{2} + \frac{1}{1} = 1,5 > 0,77$

25→ $\frac{1}{5} + \frac{1}{8} = \frac{13}{40} = 0,325 \leq 0,77$, quindi RMA

Scheduling realtime [12 punti]

Domanda 24 Considera un sistema realtime con i seguenti 2 task periodici:

- A, tempo di esecuzione 2, periodicità 7
- B, tempo di esecuzione 1, periodicità 4

Scegli l'algoritmo **più semplice/opportuno** tra i due visti durante il corso (RMA o EDF).
► **Indica chiaramente l'algoritmo scelto sulla riga tratteggiata sotto alla tabella e motiva la tua scelta su un foglio a parte. [1 punto]**
► **Completa la tabella indicando l'ordine di scheduling in base all'algoritmo scelto.** *Nota: considera che si tratta di un sistema non-preemptive con una sola CPU/un solo core.* **[5 punti]**

Tempo (ms) →	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
A	A	A						A	A						A	A						A	A						A	A			
B	B				B				B				B				B				B				B				B				B
	B	A	A		B			A	A	B			B		A	A	B				B	A	A		B				B	A	A		B

EDF ..... ☐w ☐p- ☐p+ ☒c 6/6

Domanda 25 Considera un sistema realtime con i seguenti 2 task periodici:

- A, tempo di esecuzione 2, periodicità 5
- B, tempo di esecuzione 4, periodicità 8

Scegli l'algoritmo **più semplice/opportuno** tra i due visti durante il corso (RMA o EDF).
► **Indica chiaramente l'algoritmo scelto sulla riga tratteggiata sotto alla tabella e motiva la tua scelta su un foglio a parte. [1 punto]**
► **Completa la tabella indicando l'ordine di scheduling in base all'algoritmo scelto.** *Nota: considera che si tratta di un sistema non-preemptive con una sola CPU/un solo core.* **[5 punti]**

Tempo (ms) →	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
A	A	A			A	A				A	A					A	A				A	A				A	A				A	A	
B	B	B	B					B	B	B	B					B	B	B	B					B	B	B	B					B
	A	A	B	B	B	B	A	A	B	B	B	B	A	A	/	A	A	B	B	B	B	A	A	/	B	B	B	B	A	A	A	A	B

RM ..... ☐w ☐p- ☐p+ ☒c 6/6

Scheduling non-realtime [6 punti]

▶ **Premessa alle domande che seguono** Considera un sistema che utilizza l'algoritmo di scheduling round-robin con priorità in cui vengono eseguiti i seguenti quattro processi CPU-bound. Per ogni processo il tempo di arrivo (arrivo nella coda), la sua priorità (1 = massima, 5 = minima), e il tempo stimato di esecuzione sono:

- A, arrivo dopo 2ms, priorità 4, tempo di esecuzione 9ms
- B, arrivo dopo 7ms, priorità 2, tempo di esecuzione 5ms
- C, arrivo dopo 9ms, priorità 1, tempo di esecuzione 6ms
- D, arrivo dopo 5ms, priorità 3, tempo di esecuzione 10ms

Domanda 26 Determina l'ordine di scheduling completando la tabella seguente, considerando che lo scheduler è preemptive e il **quanto dura 4ms. [4 punti]**

Tempo (ms) →	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
	/	/	A	A	A	D	D	B	B	C	C	C	C	C	C	B	B	B	D	D	D	D	D	D	D	D	A	A	A	A	A	A		

A D B C

☐0 ☐1 ☐2 ☐3 ☒4 4/4

Domanda 27 Calcola il tempo medio di elaborazione (**quanto di 3ms**) **[2 punti]**

☐w ☐p ☒c 2/2

A=30; B=11; C=6; D=21 $T = \frac{68}{4} = 17$ ms

		arrivo	esec	
1-	C	9 ms	6 ms	2
2-	B	7 ms	5 ms	3
3-	D	5 ms	10 ms	8
4-	A	2 ms	9 ms	6

quanto = 4 ms