

Sistemi Operativi 2003

Nome e cognome

Matricola

1) Quali sono le due funzioni principali di un sistema operativo?

2) Perché un sistema operativo basato su Microkernel è più robusto agli eventuali “crash” di parti del sistema?

Nome e cognome

Matricola

3) Un processore ha una *pipeline* a 4 stadi, ogni stadio richiede per l'esecuzione 1ns.
Quante istruzioni per secondo puo' eseguire questo processore?

4) Che cosa e' un *deadlock*? Spiegatelo con un esempio.

Nome e cognome

Matricola

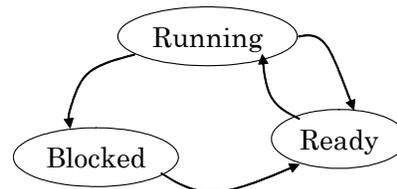
5) Descrivete la chiamata di sistema `fork()`. Che cosa succede dentro il sistema operativo quando viene chiamata `fork()`? Che cosa succede al processo chiamante?

6) Che cosa e' il *multiprogramming*? Come funziona?

Nome e cognome

Matricola

7) Perché potrebbe fallire la chiamata `fork()`?



8) Nel grafo di transizione degli stati di un processo (si veda sopra) ci sono solamente 4 possibili transizioni. Ci sono dei casi nei quali potrebbero esistere anche le transizioni mancanti (`blocked → running` o `ready → blocked`)? Giustificare la risposta.

Nome e cognome

Matricola

9) Che cosa sono i *thread*? Perché può essere utile scrivere delle applicazioni *multithread*?

10) Confrontate l'implementazione dei *thread* in *user space* con quella in *kernel space*.

Nome e cognome

Matricola

11) Perché è più facile programmare se possiamo utilizzare delle chiamate di I/O bloccanti?

12) Che cosa è un semaforo?

Nome e cognome

Matricola

13) Che cosa e' una *race condition*? Potete fare un esempio di *race condition*?

14) Che cosa e' l'istruzione TSL? Come funziona?

Nome e cognome

Matricola

15) Che cosa e' un *mutex*?

16) Che cosa e' uno *scheduler*? Di che cosa si occupa lo scheduler nel sistema operativo?

Nome e cognome

Matricola

17) Che cosa vuol dire *compute bound* o *I/O bound*? Che cosa ci consente di fare il sapere che un processo e' compute o I/O bound?

18) Che cosa vuol dire *preemptive*? Se trovaste scritto che un sistema operativo e' preemptive che cosa vorrebbe dire?

Nome e cognome

Matricola

19) Perché decidere quanto e' lungo il quanto di tempo assegnato ad un processo in un sistema *preemptive* e' un fattore chiave nel progetto del sistema operativo?

20) Descrivere che cosa succede quando viene eseguita una chiamata di sistema?

Nome e cognome

Matricola

21) Se un processo multithread fa una fork c'è un problema nel caso in cui il processo figlio riceva anche la copia di tutti i thread del processo genitore. Immaginiamo che uno dei thread del processo padre fosse blocked in attesa di input da tastiera. Avremmo a questo punto 2 thread in attesa dello stesso input da tastiera, uno per ogni processo. E' possibile avere lo stesso problema in processi single-thread?

22) Che cosa è il busy-waiting? Si faccia un esempio.

Nome e cognome

Matricola

23) Che cosa contiene (in generale) ogni entry della tabella dei processi di un sistema operativo?

24) La chiamata

`count = read (fd, buffer, nbytes)`

puo' restituire `count != nbytes`? Se si in quali casi.