

Sistemi Operativi L-A
CdS Ingegneria Informatica - Professor Paolo Bellavista
Appello di martedì 26 giugno 2007

Compito A - Parte di Teoria (8 punti)

1) Si descriva la sequenza di operazioni tipicamente svolte dal sistema operativo all'occorrenza di un page fault, in un sistema con memoria paginata e non segmentata. Si presentino, inoltre, le principali motivazioni alla base di page fault in un sistema con memoria virtuale.

2) Si illustri come opera l'algoritmo di scheduling SJF, quali sono le sue caratteristiche principali, le proprietà che permette di ottenere per il comportamento globale del sistema, e i principali problemi associati alla sua implementazione in un sistema operativo multiprogrammato interattivo.

3) Si mostri lo pseudo-codice che risolve il problema della sincronizzazione di task nell'accesso ad una risorsa con "mutua esclusione a coppie". Ovvero, deve essere consentito l'accesso alla risorsa solo a due task per volta: mai più di due task possono avere accesso concorrente alla risorsa, né mai un task da solo deve poter accedere a tale risorsa. Si definisca una soluzione per tale problema di sincronizzazione attraverso l'uso opportuno di **wait()** e **signal()** di Dijkstra.

Si rammenta ai candidati che durante la prova scritta non può essere consultato nessun tipo di materiale (slide del corso, appunti, libri, ...); l'accesso al sito Web del corso è stato disabilitato.

Sistemi Operativi L-A
CdS Ingegneria Informatica - Professor Paolo Bellavista
Appello di martedì 26 giugno 2007

Compito B - Parte di Teoria (8 punti)

1) Si descrivano precisamente i meccanismi di sistema operativo messi in azione dall'invocazione ed esecuzione di una system call. Quali sono le principali motivazioni per l'introduzione e il supporto di system call in ogni sistema operativo moderno?

2) Si illustri nel dettaglio il meccanismo di allocazione a indice per i blocchi di un dispositivo di memoria di massa. Quali specifici vantaggi ha questa strategia di allocazione rispetto a soluzioni basate su allocazione contigua o su allocazione a lista?

3) Si mostri, tramite un piccolo esempio completo di codice Java, come sia possibile istanziare e utilizzare thread Java senza fare sub-classing della classe di sistema Thread. Si descrivano inoltre le motivazioni dell'esistenza di tale possibilità, eventualmente aiutandosi con esemplificazioni anche in questo caso.

Si rammenta ai candidati che durante la prova scritta non può essere consultato nessun tipo di materiale (slide del corso, appunti, libri, ...); l'accesso al sito Web del corso è stato disabilitato.