

# SISTEMI OPERATIVI - IIN

Ing. Pierfrancesco Bellini, A.A. 2015-16

## 1 INTRODUZIONE

### 1.1 INTRODUZIONE AI SISTEMI OPERATIVI ([1] CAP. 1)

Cosa è un sistema operativo, punto di vista utente, punto di vista del sistema. Organizzazione di un sistema di calcolo, funzionamento, struttura memoria, struttura I/O. Struttura del sistema operativo. Attività del sistema operativo, modalità utente e sistema, timer.

### 1.2 STRUTTURA DI UN SISTEMA OPERATIVO ([1] CAP. 2)

Servizi di un sistema operativo. Chiamate di sistema. Struttura del sistema operativo: struttura semplice, metodo stratificato, microkernel, moduli. Macchine virtuali (cenni).

## 2 LE BASI DEL LINGUAGGIO JAVA

### 2.1 JAVA

La tecnologia Java e la Java Virtual Machine. Classi ed Oggetti. Tipi di dati. Passaggio dei parametri. Array. Package. Gestione delle eccezioni. Classi derivate. Interfacce. Polimorfismo.

## 3 GESTIONE DEI PROCESSI

### 3.1 I PROCESSI ([1] CAP. 3)

Concetto di processo. Scheduling dei processi. Operazioni sui processi. Comunicazione tra processi: memoria condivisa e scambio messaggi. I processi in Linux, le system call getpid(), getppid(), fork(), exec(), wait() e exit(). Esempi in linguaggio C. Il controllo dei processi in Java: principali classi e metodi. Esempi.

### 3.2 I THREAD ([1] CAP. 4)

Introduzione. Modelli di programmazione multi-thread. Librerie di thread: pthreads, threads in Java. La classe Thread e l'interfaccia Runnable. I principali metodi della classe Thread. Stati e scheduling dei thread Java. Esempi.

### 3.3 SCHEDULING DELLA CPU ([1] CAP. 5)

Concetti fondamentali: CPU burst, I/O burst, scheduling con diritto di prelazione, dispatcher. Criteri di scheduling. Algoritmi di scheduling: scheduling in ordine di arrivo, scheduling per brevità, scheduling per priorità, scheduling circolare, scheduling a code multiple, scheduling a code multiple con retroazione. Scheduling dei thread. Scheduling di linux: kernel 2.4.x e kernel 2.6.x. Completely fair scheduling.

### 3.4 SINCRONIZZAZIONE TRA PROCESSI ([1] CAP. 6)

Introduzione. Problema della sezione critica. Soluzione di Peterson per due processi. Soluzione per N processi (algoritmo di Lamport). Hardware per la sincronizzazione. Semafori. Stallo e attesa indefinita. Problemi tipici di sincronizzazione. Produttori e consumatori con memoria limitata. Problema dei lettori-scrittori. Problema dei 5 filosofi. I monitor. Soluzione al problema dei filosofi con uso di monitor. Sincronizzazione in Java: uso di metodi synchronized, metodi wait(), notify() e notifyAll(). Semafori e monitor in Java. Sincronizzazione di blocco.

### 3.5 STALLO ([1] CAP. 7)

Modello dei sistema. Caratterizzazione delle situazioni di stallo, condizioni necessarie. Grafo di assegnazione delle risorse. Prevenire situazioni di stallo. Evitare le situazioni di stallo, stato sicuro, algoritmo con grafo di assegnazione delle risorse, algoritmo del banchiere. Rilevamento delle situazioni di stallo. Ripristino da situazioni di stallo.

## 4 GESTIONE DELLA MEMORIA

### 4.1 GESTIONE DELLA MEMORIA CENTRALE ([1] CAP. 8)

Introduzione, dispositivi essenziali, associazione degli indirizzi, spazi di indirizzi logici e fisici. Allocazione contigua della memoria, rilocazione e protezione della memoria, allocazione della memoria, frammentazione. Paginazione, metodo di base, architettura di paginazione, protezione, pagine condivise. Struttura delle tabelle delle pagine, paginazione gerarchica, tabelle delle pagine di tipo hash, tabelle delle pagine invertita. Segmentazione, metodo di base, architettura di segmentazione.

### 4.2 MEMORIA VIRTUALE ([1] CAP. 9)

Introduzione. Paginazione su richiesta. Prestazioni della paginazione su richiesta. Sostituzione delle pagine, schema di base, FIFO, ottimale e LRU.

## 5 BIBLIOGRAFIA

[1] *Sistemi Operativi: Concetti ed esempi*, ottava edizione, Pearson Addison-Wesley, A. Silberschaz, P. Galvin, G. Gagne.