

Sistemi operativi e macchine virtuali

Lezione 9 di Fondamenti di informatica

Docenti: Marina Madonia & Giuseppe Scollo

Università di Catania

Facoltà di Scienze Matematiche, Fisiche e Naturali
Corso di Laurea in Informatica, I livello, AA 2008-09

Indice

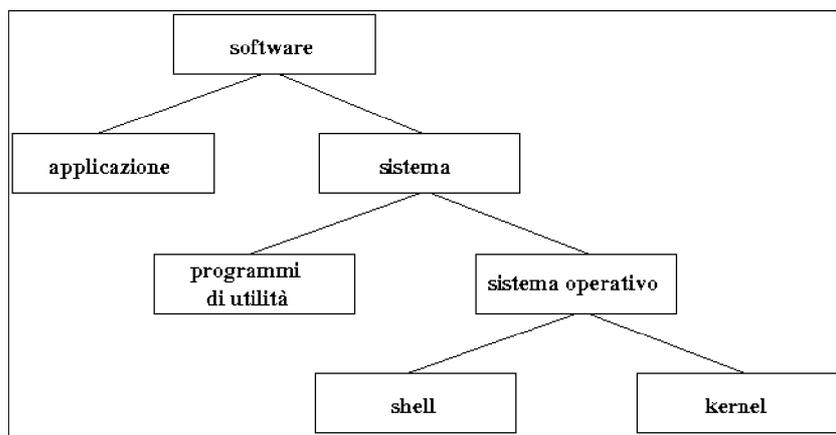
1. Sistemi operativi e macchine virtuali
2. evoluzione dei sistemi operativi
3. tassonomia del software
4. componenti del sistema operativo
5. avvio del sistema operativo
6. processi in *time-sharing*
7. gestione di processi e risorse
8. modello *client-server*
9. esercizi e approfondimenti

evoluzione dei sistemi operativi

- elaborazione a lotti** (ingl. *batch processing*): i primordi
gli utenti sottopongono lotti di *job* per l'esecuzione, serviti con disciplina FIFO
ciascun *job* consta di: programma, dati, istruzioni JCL (v.appresso)
l'utente ottiene i risultati solo dopo la fine dell'esecuzione del *job*
Job Control Language (JCL): istruzioni premesse al programma, per:
il sistema operativo (max. memoria richiesta, tempo di CPU, ...)
l'operatore (umano), ad es.: montaggio di nastri
l'utente non ha accesso alla macchina (direttamente)
- elaborazione interattiva**: a colloquio con la macchina
l'utente interagisce con il programma in esecuzione e con il sistema (JCL → *shell*)
l'utente ha accesso alla macchina (da terminale video)
operatore → amministratore di sistema
- elaborazione distribuita**: le rivoluzioni della Rete
reti di calcolatori: omogenee (proprietarie, chiuse), eterogenee (aperte: protocolli standard)
inter-reti: Internet, la Rete delle reti
elaborazione parallela: sistemi multiprocessore
elaborazione cooperativa in Rete: GRID computing
problemi comuni: bilanciamento del carico, scalabilità, sicurezza

tassonomia del software

una classificazione approssimativa, "dinamica"



tendenza alla "migrazione in discesa" del software
da "applicazione" a "sistema"

componenti del sistema operativo

gli "immancabili":

kernel ("nocciolo"), comprende almeno:

memory manager : gestore della memoria

scheduler : pianificatore dell'esecuzione

dispatcher : smistatore delle richieste di esecuzione

device drivers : interfacce SW ai dispositivi HW

file manager : gestore dei file

shell : interprete del linguaggio di sistema

window manager : gestore delle finestre ("immancabile"?)

avvio del sistema operativo

il problema del *bootstrap* :

un problema ricorrente (a cominciare dal ... Big Bang!)

in due passi:

1. all'avvio si esegue il programma di *bootstrap*, residente in ROM
il sistema operativo risiede in memoria di massa
2. il programma di *bootstrap* trasferisce il sistema operativo nella memoria principale, e poi gli cede il controllo

nei sistemi odierni possono aversi:

più dispositivi di avvio (disco, floppy, CD/DVD, USB drive, rete)

avvio di uno fra più sistemi operativi (*boot loader*)

bootstrap a più stadi

processi in *time-sharing*

l'idea del *time-sharing* :

sfruttare i tempi di attesa di I/O per l'esecuzione di altri programmi

le risorse di calcolo sono allocate a turno a **processi** che ne fanno richiesta

ad ogni processo in esecuzione è assegnato un **quanto** di tempo massimo per l'uso ininterrotto della CPU

la **commutazione di processo** ha luogo quando il processo in esecuzione:

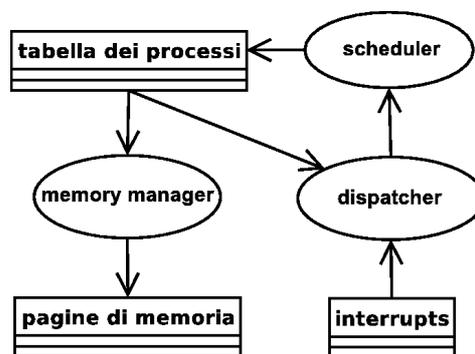
esaurisce il quanto di tempo assegnato, oppure

effettua una richiesta di I/O

gestione di processi e risorse

una mediazione continua ...

scheduler, dispatcher, memory manager

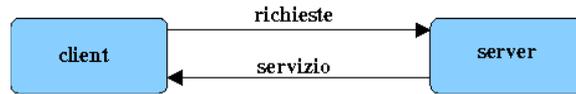


schema dell'allocazione di risorse ai processi

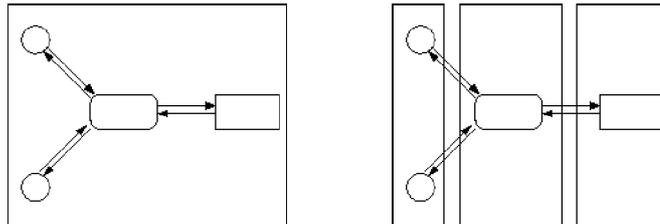
modello *client-server*

un modello ricorrente:

reinventato nel concetto di **oggetto software** (1967)



acquista nuovi significati nell'**elaborazione distribuita**:



esercizi e approfondimenti

1. Nella commutazione di processo in un sistema *time-sharing*, quale dei seguenti componenti del *kernel* determina l'allocazione della CPU al prossimo processo da eseguire? Quale di essi aggiorna la tabella dei processi residenti in memoria?
(a) *interrupt handler*, (b) *scheduler*, (c) *dispatcher*, (d) *memory manager*
2. Le architetture software *multi-tier* generalizzano il modello *client-server* prevedendo che un componente software possa essere al contempo *server* per l'esecuzione di una data richiesta di servizio e *client* nell'interazione con altri *server*, per richieste di servizio la cui esecuzione sia necessaria a soddisfare la prima. Trovare un esempio di servizi, tratto dalla vita quotidiana, in cui si manifesti questa situazione, dove i ruoli di *client* e *server* siano giocati da persone o organizzazioni.
3. La sicurezza del sistema operativo è un prerequisito essenziale alla sicurezza delle risorse (hardware, software, informazione) che esso gestisce e dei servizi che fornisce. Riservatezza, integrità e disponibilità sono le principali caratteristiche della sicurezza di un sistema informatico. Si può approfondire il tema della sicurezza nei sistemi operativi con riferimento alle molteplici funzioni che essi realizzano al fine di garantire le suddette caratteristiche.