

3 - Componenti

sabato 26 marzo 2022 11:54

GESTIONE DEI PROCESSI

- **Processo:** programma in esecuzione
 - necessità di risorse
 - eseguito in modo sequenziale un'istruzione alla volta
- **Il S.O. si occupa di:**
 - creare e distribuire processi
 - sospendere e riprendere processi
 - sincronizzare e far comunicare processi tra loro

GESTIONE DELLA MEMORIA PRIMARIA (RAM)

- **RAM:** conserva dati condivisi dalla CPU e dai dispositivi I/O
 - un programma deve essere caricato in memoria per poter essere eseguito
 - piccola e volatile
- **Il S.O. si occupa di:**
 - gestire lo spazio della memoria
 - decidere quale processo caricare in memoria quando c'è spazio disponibile
 - allocare e rilasciare spazio di memoria

GESTIONE DELLA MEMORIA SECONDARIA (es. Hard Disk)

- Utile per mantenere grandi quantità di dati in modo permanente
- Tipicamente uno o più dischi magnetici
- **Il S.O. si occupa di:**
 - gestire lo spazio libero su disco
 - allocare spazio su disco
 - organizza gli accessi su disco

GESTIONE DELL'I/O

- Il S.O. nasconde all'utente le caratteristiche dei dispositivi I/O
- Il sistema di I/O comprende:
 - sistema per accumulare gli accessi ai dispositivi (buffering)
 - interfaccia verso i device driver
 - device driver specifici per alcuni dispositivi

GESTIONE DEI FILE

- **File:** raccolta di informazioni correlate (dati o programmi)
- I file vengono memorizzati su supporti fisici (dischi, DVD, SD, ecc.) e controllati da driver con caratteristiche diverse
- **Il S.O. si occupa di:**
 - creare e cancellare file e cartelle
 - supporto per la gestione di file e cartelle (copia, sposta, modifica, ...)
 - salvare le informazioni a scopo di backup

PROTEZIONE

- Per protezione si intende il meccanismo con cui si controlla l'accesso alle risorse da parte di utenti e processi
- **Il S.O. si occupa di:**
 - definire gli accessi autorizzati e non (autorizzare accessi e bloccarli)
 - fornire strumenti per verificare la politica di accesso

SISTEMI DISTRIBUITI

- **Sistema distribuito:** tipologia di sistema informatico costituito da un insieme di processi interconnessi tra loro
 - risorse di calcolo connesse tramite una rete
 - non condividono nè la memoria nè un clock

- **Il S.O. si occupa di gestire "in rete" le varie componenti:**

- processi distribuiti
- memoria distribuita
- file system distribuiti
- ...

SYSTEM CALLS

- **System Call:** meccanismo usato a livello utente per richiedere un servizio a livello kernel
 - permettono di eseguire operazioni riservate quali l'accesso all'hardware o alla memoria
 - ESEMPLI: open(), read(), write(), wait(), ...

INTERPRETE DEI COMANDI - SHELL

- S.O. fornisce all'utente un **interprete di comandi** (detto anche **shell**)
 - legge i comandi dell'utente e li esegue (esegue istruzioni)
- L'interprete fornisce anche diversi tipi di interfaccia
 - CLI: Command-Line
 - GUI: Graphics User Interface
 - Batch

CLI: COMMAND-LINE

- Permette di digitare direttamente i comandi (es. UNIX)
 - Spesso implementata nel kernel
 - Spesso implementata in diversi varianti - shells

GUI: GRAPHICS USER INTERFACE

- Tutto viene rappresentato da icone (file, programmi, azioni, ecc)
- Tutto è interagibile attraverso mouse e tastiera

API

- **API:** applicazioni il cui software permette a 2 applicazioni di comunicare tra loro
 - Le API fanno uso di diverse system calls
 - ESEMPIO: l'app del meteo sul nostro cellulare è una API, mette in comunicazione il nostro cellulare con i server del meteo
- Le due API più comuni:
 - Win32 API (insieme di API usate su Windows)
 - POSIX (insieme di API usate su tutte le versioni di UNIX, Linux e Mac OS X)

USARE LE SYSTEM CALLS

- Viene associato un numero ad ogni system call
 - l'interfaccia alle chiamate ha una tabella che associa del codice (system call) ad ogni numero
 - l'interfaccia invoca la system call nel kernel del SO

- 1) System call invocata dal programma utente
- 2) Si passa da modalità utente a modalità kernel
- 3) Viene utilizzata la tabella per chiamare la system call corretta
- 4) Si passa da modalità kernel a modalità utente ritornando il risultato

COMUNICAZIONE TRA SO E UN PROCESSO

- 3 opzioni:
 - passare parametri (dalla system call) tramite registri
 - passare parametri tramite stack
 - salvare i parametri in una tabella di memoria (l'indirizzo di tabella viene poi passato in un registro o stack)

PASSAGGIO DI PARAMETRI NELLO STACK

- Lo stack è FIFO (First In First Out) quindi i parametri vengono salvati uno sopra l'altro
- Quando non servono più, questi NON VENGONO CANCELLATI (sarebbe uno spreco di tempo)
 - vengono sovrascritti da nuovi parametri

