Laboratorio di Sistemi Operativi

primavera 2009

Introduzione

Introduzione

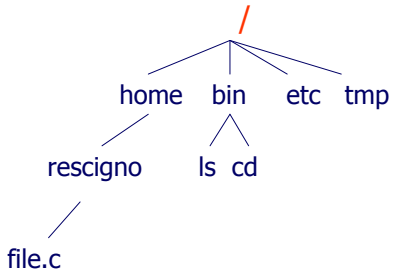
- Ogni s.o. fornisce servizi ai programmi che girano
  - esecuzione di nuovi programmi
  - aprire/chiudere file
  - allocare memoria
- Noi studieremo i servizi offerti da UNIX
- Descrivere un sistema operativo è impossibile se non attraverso l'introduzione ed una panoramica dei suoi "termini" salienti

Introduzione

- shell: interprete di comandi
  - sh (include reminiscenze di ALGOL)
  - csh (usa una sintassi simile al C, prende comandi interattivi o da programmi)
  - tcsh (come la csh + permette di viaggiare su e giù per la lista dei comandi dati, spelling correction dei comandi)
  - bash

Introduzione

- Filesytem gerarchico
  - files & directory
  - root **/**



- Pathname: sequenza di zero o più nomi di file separati da /

Es: /home/rescigno/file.c

- Nomi di file speciali:
  - dot **.**
  - dot dot **..**

Introduzione

- "File eseguibile":
  - letto dalla memoria
  - eseguito dal kernel
- Processo: istanza in esecuzione di un "eseguibile" (detto anche task)
- Ogni processo ha un identificatore numerico pid>0 associato mediante il quale ci si riferisce

Errori

- quando avviene un errore in una funzione UNIX, viene restituito un intero negativo (-1, in genere)
- la vbl intera errno è settata ad un valore che fornisce ulteriori info
- vedi la sezione 2 del manuale di UNIX: man 2 intro
- <errno.h> contiene i valori costanti assumibili da errno
  - In linux: </usr/include/errno.h>

Identificatori utenti

- Un utente viene identificato dal sistema attraverso:
  - User Id : valore numerico che ci identifica in /etc/passwd
  - Group Id : nel file /etc/group c'è la mappa tra i nomi dei gruppi ed i rispettivi identificatori numerici
- superuser ha uid = 0

Esempi:

ls –l usa /etc/passwd per printare la login name del proprietario

```
printf("uid =%d, gid =%d\n",getuid(),getgid());
```


System Call & librerie

- Una system call è un entry point del kernel per fornire servizi ai processi che li richiedono
- man 2 fornisce la documentazione sulle system call (definite in C)
- system call → funzione omonima nella libreria standard (wrapper)
  - Es. ssize_t read (int *filedes*, void **buff*, size_t *nbytes*);
- l'utente chiama il wrapper (attraverso la sequenza standard di chiamate a funzioni di C), questo invoca il servizio del kernel
- Semplifichiamo: System Call = Funzioni C

System Call & Librerie

- man 3 contiene le funzioni general purpose per il programmatore (libc)
  - printf, malloc, etc.
  - non sono system call

Esempio:

il codice vuole allocare memoria → invoca malloc la quale per allocare realmente utilizza la system call sbrk

System Call & Librerie

- system call: interfaccia minima, mentre le funzioni di libreria forniscono più elaborate funzionalità
- libc: interfaccia normale
- system call sono definitive dal s.o., le funzioni di libreria no!