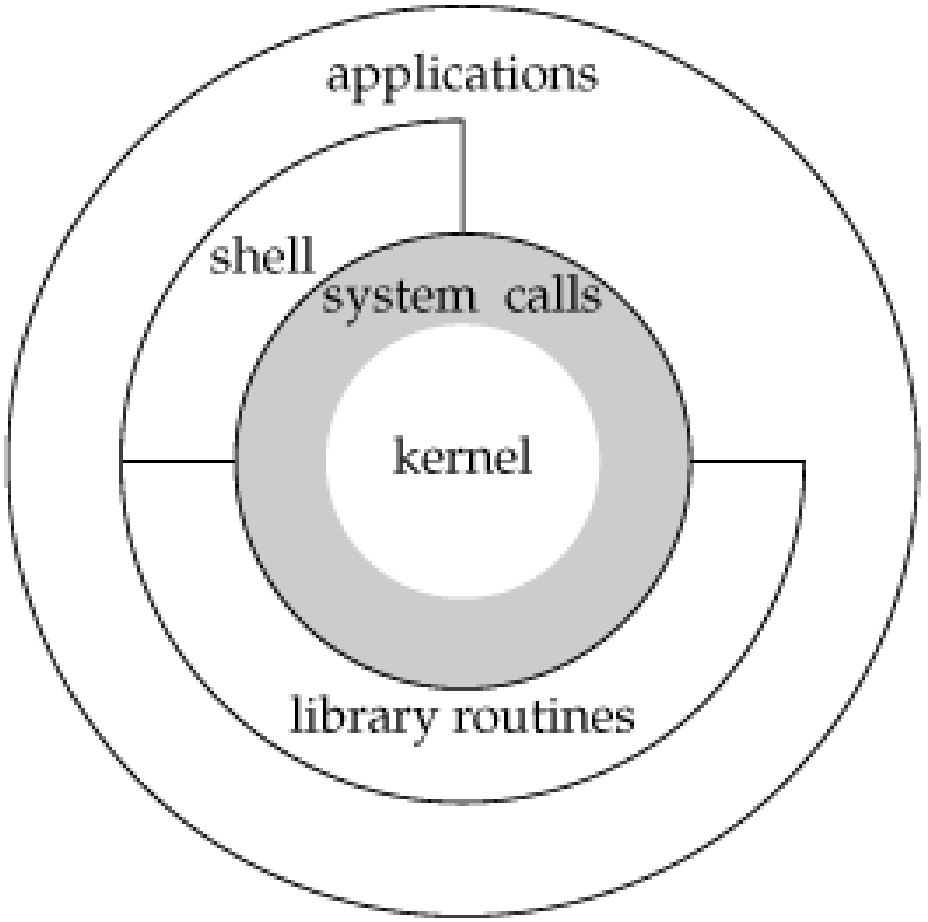
System call



- Tutte le implementazioni del sistema operativo UNIX forniscono un insieme limitato e ben definito di punti di acceso al kernel chiamate system call
- Le system call sono accessibili tramite funzioni della liberia standard C

system call e funzioni di libreria

- Dal punto di vista del programmatore le system call sono funzioni C
- L'implementazione di una system call è completamente trasparente all'utente e prevede l'interazione con l'hardware attraverso il kernel
- Quando il programma invoca una system call, il kernel interagisce con l'hardware del sistema fornendo i servizi richiesti

system call VS library function

- Si consideri il problema dell'allocazione della memoria: esistono varie tecniche per l'allocazione della memoria e la liberazione delle aree inutilizzate (garbage collection)
- La system call sbrk semplicemente aumenta o diminuisce lo spazio di indirizzamento di un processo del numero di byte specificato delegando la gestione dello spazio allocato al programma
- La funzione di memoria malloc implementa una particolare politica di gestione della memoria allocata ed utilizza la funzione sbrk
- C'è una chiara separazione dei ruoli: la system call alloca lo spazio necessario, la funzione di libreria lo gestisce per l'utente

system call VS library function

- Si consideri il problema della determinazione dell'orario
- Alcuni sistemi operativi che forniscono una system call che restituisce l'orario corrente e quindi gestiscono avvenimenti occasionali quali il passaggio tra ora legale e solare tramite kernel
- I sistemi UNIX forniscono una sola system call (time) che restituisce il numero di secondi passati dalla mezzanotte del 1° Gennaio 1970 a single system call that returns the number of seconds since the Epoch: midnight, January 1, 1970, UTC
- Ogni interpretazione di questo numero come ad esempio la conversione nel formato d'uso quotidiano

  ore:minuti giorno/mese/anno

  è lasciato al processo utente
- La libreria standard C fonisce varie funzioni per l'interpretazione e la conversione di questo valore

L'interpretazione dell'orario

- Il comando date interpreta l'informazione fornita dal kernel visualizzando l'ora corrente in formato leggibile
- Per stabilire l'ora ha bisogno di sapere anche il fuso orario in cui si trova l'utente che effettua il comando
- Sul sistema è impostato un fuso orario di default che è normalmente quello dell'area geografica in cui si trova
- Se l'utente vuole modificare il fuso orario con cui gli orari vengono interpretati può settare la variabile TZ
- Esempio:
  $ date
  Wed May 11 08:24:22 CEST 2011
  $ export TZ=America/New_York
  $ date
  Wed May 11 02:24:22 EDT 2011
- Non c'e' bisogno di modificare l'orario del sistema
- La variabile agisce anche sulle funzioni di libreria ctime, localtime, ...

system call VS library function

- Una differenza sostanziale tra funzioni di libreria e system call è che le prime forniscono funzionalità elaborate, mentre le ultime hanno solitamente una interfaccia molto essenziale

Il comando man

- Le pagine di manuale di un sistema UNIX sono raggruppate in 8 sezioni:

1) programmi eseguibili o comandi di shell (ls, vi, ...)

2) system call (open, write, fork, ...)

3) funzioni di libreria (printf, scanf, ...)

4) file speciali (/dev/null, /dev/mouse, ...)

5) formati di file (/etc/passwd, /etc/hosts, ...)

6) Giochi (fortune, ...)

7) Protocolli, convenzioni, panoramiche, varie (tcp, boot, ...)

8) Comandi per la gestione del sistema (iptables, mkfs, ...)

Selezionare una sezione

- Le pagine di manuale vengono ricercate all'interno delle sezioni in ordine crescente
- Alla prima corrispondenza trovata viene visualizzata la pagina di manuale individuata
- Alcune parole chiave possono essere presenti in più sezioni quindi per individuare quella specifica desiderata dobbiamo preporre il numero della sezione
- La parola write corrisponde almeno a due pagine di manuale:
  - Il comando write per comunicare con gli utenti
  - La system call write per l'I/O a basso livello
- Per visualizzare la pagina della system call usiamo:

  $ man 2 write

Sottosezioni di una pagina man

- All'interno di una pagina di manuale ci sono varie sottosezioni
- Quelle convenzionali sono NAME, SYNOPSIS, CONFIGURATION, DESCRIPTION, OPTIONS, EXIT STATUS, RETURN VALUE, ERRORS, ENVIRONMENT, FILES, VERSIONS, CONFORMING TO, NOTES, BUGS, EXAMPLE, AUTHORS, SEE ALSO.
- Particolare rilevanza nelle pagine di manuale delle system call sono le sottosezioni evidenziate in rosso

man 2 write

NAME

write - write to a file descriptor

SYNOPSIS

```
#include <unistd.h>

ssize_t write(int fd, const void *buf, size_t count);
```

DESCRIPTION

write() writes up to count bytes from the buffer pointed buf to the file referred to by the file descriptor fd …

RETURN VALUE

On success, the number of bytes written is returned (zero indicates nothing was written). On error, -1 is returned, and errno is set appropriately...

ERRORS

EAGAIN The file descriptor fd refers to a file other than a socket and has been marked nonblocking (O_NONBLOCK), and the write would block...

CONFORMING TO

SVr4, 4.3BSD, POSIX.1-2001....

NOTES

A successful return from write() does not make any guarantee that data has been committed to disk...

SEE ALSO

close(2), fcntl(2), fsync(2), ioctl(2), lseek(2), open(2), pwrite(2), read(2), select(2), writev(2), fwrite(3)