SC - 27/5/2025

27/05/25 10:09

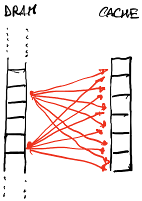
Esempio di uso cache (4 linee da 64 byte ciascuna)

Indirizzo acceduto	numero blocco	Contenuto cache				
		64b				
20	0	0(0)				COLD CACHE MISS
		0	1	2	3	
464	7	0(1)	7(0)			" " "
128	2	0(2)	7(1)	2(0)		" " "
36	0	0(0)	7(2)	2(1)		CACHE HIT
608	9	0(1)	7(3)	2(2)	9(0)	COLD CACHE MISS
360	5	0(2)	5(0)	2(3)	9(1)	CAPACITY CACHE MISS
204	3	0(3)	5(1)	3(0)	9(2)	" " "

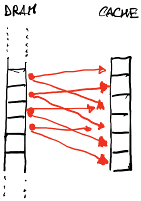
t

ASSOCIATIVITÀ

CACHE COMPLETAMENTE ASSOCIATIVA

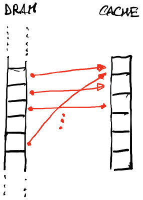


CACHE ASSOCIATIVA A K-VIE



Esempio K=2

CACHE CON MAPPING DIRETTO



A seconda di quale tipo di cache adottiamo abbiamo diversi risultati per ogni accesso in memoria.

CACHE HIT = il dato è presente in cache

COLD CACHE MISS : " " non è presente in cache perché la cache è vuota

CAPACITY CACHE MISS : " " " " " " " e la cache è piena

CONFLICT CACHE MISS : " " " " " " " e confligge con altri dati presenti nelle K-vie.

Es.: 23/7/19 - A q. 2.

K=2 line: 4 dim = 32 byte Accessi: 105, 782, 384, 1356, 11, 324, 265

ADDR	BLOCK	CACHE				
		0	1	2	3	
105	3	–	–	3(0)	–	COLD

782	24	24(0)	—	3(1)	—	//
384	12	24(1)	12(0)	3(2)	—	//
1356	42	42(0)	12(1)	3(3)	—	CONFLICT
11	0	42(1)	0(0)	3(4)	—	//
324	10	10(0)	0(1)	3(5)	—	//
265	8	10(1)	8(0)	3(6)	—	//

risultato : 10, 8, 3, –

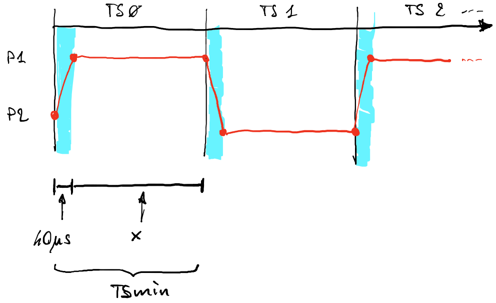
Es. 25/7/19 q. 1

$A \rightarrow K = 3,5x \qquad P \rightarrow 1,4x = S \qquad S = \dfrac{1}{\frac{\alpha}{K} + 1 - \alpha}$

$$1,4 = \frac{1}{\frac{\alpha}{3,5} + 1 - \alpha} = \frac{3,5}{3,5 - 2,5\,\alpha} =$$

$$2,5\,\alpha = 3,5 - \frac{3,5}{1,4} \rightarrow \alpha = 0,4 = 40\%$$

Es. 25/7/19 q. 3


TS 0
TS 1
TS 2 ---
P1
P2
40μs
x
TSmin

$$40 = 0,02 * TSmin = 0,02 * (40 + x)$$

$$x = \frac{40}{0,02} - 40 = 1,96 \text{ msec}$$

Esercizio 25/7/19 q 4

indirizzi a 20 bit → spazio indirizzabile 2^{20}

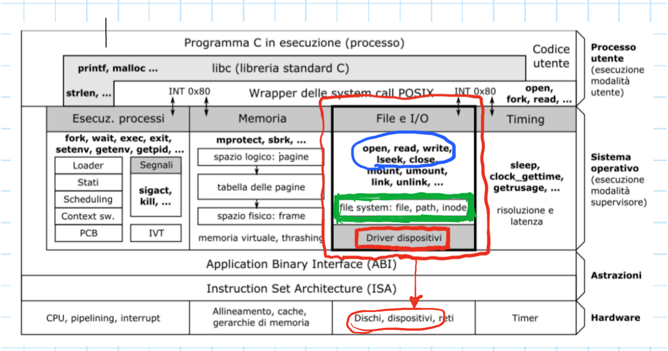
Tabella pagine 2KB ogni entry 4 byte

2KB = 2048 byte = 2^{11} num righe = $2^{11}/2^2 = 2^9$

dim. pagina = dim RAM / num_pagine = $2^{20}/2^9 = 2^{11}$ byte = 2KB

INTRODUZIONE AL FILE SYSTEM

E l'insieme di schemi organizzativi dei dati e le relative funzionalità attraverso cui il SO si interfaccia con le periferiche di archiviazione dati


Programma C in esecuzione (processo)
Codice utente
printf, malloc ...
libc (libreria standard C)
strlen, ...
INT 0x80
Wrapper delle system call POSIX
INT 0x80
open, fork, read, ...
Processo utente (esecuzione modalità utente)
Esecuz. processi
fork, wait, exec, exit, setenv, getenv, getpid, ...
Loader
Stati
Scheduling
Context sw.
PCB
Segnali
sigact, kill, ...
IVT
Memoria
mprotect, sbrk, ...
spazio logico: pagine
tabella delle pagine
spazio fisico: frame
memoria virtuale, thrashing
File e I/O
open, read, write, lseek, close, mount, umount, link, unlink, ...
file system: file, path, inode
Driver dispositivi
Timing
sleep, clock_gettime, getrusage, ...
risoluzione e latenza
Sistema operativo (esecuzione modalità supervisore)
Application Binary Interface (ABI)
Instruction Set Architecture (ISA)
Astrazioni
CPU, pipelining, interrupt
Allineamento, cache, gerarchie di memoria
Dischi, dispositivi, reti
Timer
Hardware

FILE E DIRECTORY

FILE : contenitori di dati

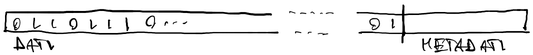
DIRECTORY : contenitori logici che permettono l'organizzazione dei dati in una struttura ad albero.

- i nodi sono directory
- le foglie = file

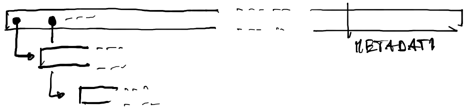
FILE : archivio di dati rappresentati come sequenza di byte. Accessibile in lettura/scrittura. Oltre ai dati il file contiene una serie di metadati

- nome
- dimensione
- permessi di accesso

...

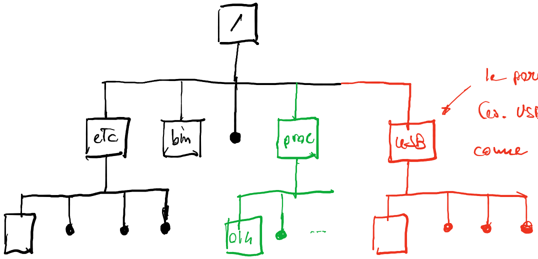

0 1 1 0 1 1 1 0 ...
----
0 1
DATI
METADATI

DIRECTORY : contenitore logico di altri file e directory



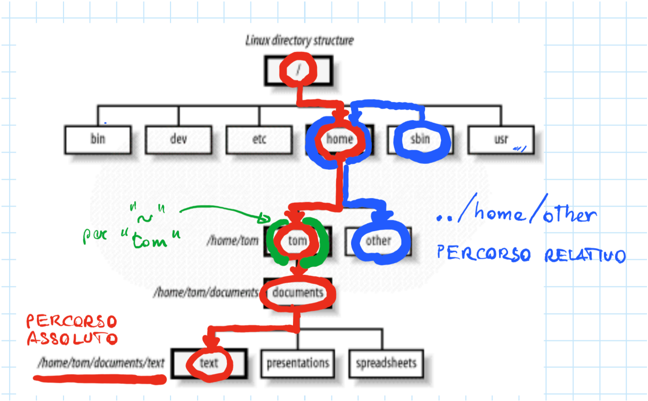
STRUTTURA GERARCHICA DEL FILE SYSTEM

Nel mondo UNIX il contenuto del file system è organizzato in un unico albero. È presente una directory speciale chiamata "root directory" e rappresentata "/"


/
etc
bin
proc
USB
014
---

le periferiche esterne (es. USB Key) sono aggiunte come sottoalberi attraverso il comando "mount"

FILE SYSTEM HIERARCHY STANDARD (FSHS)

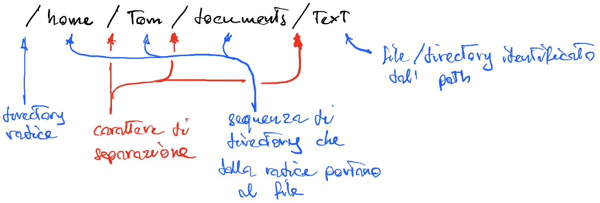

Linux directory structure
/
bin
dev
etc
home
sbin
usr
/home/tom
tom
other
/home/tom/documents
documents
/home/tom/documents/text
text
presentations
spreadsheets
per "tom" "~"
../home/other
PERCORSO RELATIVO
PERCORSO ASSOLUTO

PATH / PERCORSO

Nel file system ogni file / directory è identificato univocamente dal suo "path"

PATH ASSOLUTO: sequenza di tutte le directory presenti nel ramo del FS che partendo dalla radice arriva fino al file/dir considerato

Es.:


/home/Tom/documents/Text
directory radice
carattere di separazione
sequenza di directory che dalla radice portano al file
file/directory identificato dal path

ATTENZIONE: "/" è un path valido

"//" non è un path valido

PATH RELATIVO: sequenza di directory presenti nei rami del FS che collegano il file/dir considerato con una directory di riferimento passando per la dir "antenato" comune ad entrambi i rami

Per definire un path relativo usiamo i percorsi speciali:

"." indica la directory corrente

".." " " " padre di quella corrente

"~" " " " home dell'utente

Es. per indirizzare "other" partendo da "documents":

../../other