

VECTORES

* Solo necesitas saber la fórmula para obtener la dirección de un elemento de v .

$$V[i] = \&V + i \cdot \text{Tam}$$

↪ Tam es el tamaño del vector (.word, .short,...)

ej:

0	1	2	3	4	5	6	7
3	0	-1	4	8	6	-2	0

$V[3] : \text{ . word } 3, 0, -1, 4, 4, 6, -2, 0$

$V[0] = \&V \rightarrow$ primera pos.

Si vull anar a $V[4] : V[4] = \&V + 4 \cdot i = \&V + 4 \cdot 4 = \&V + 16$.

Si ara volgués accedir a la info de $V[4] :$

la $\$t0, v \quad \# \$t0 = \&V$

addu $\$t1, \$t0, 16 \quad \# \$t1 = @V[4]$

lw $\$t1, 0(\$t1) \quad \# \$t1 = 8$

MATRICES

* Las matrices se guardan de izquierda a derecha por filas

* Tipos de accesos:

↪ num columns

$$M[i][j] = \&M + i \cdot \text{NC} \cdot \text{Tam} + j \cdot \text{Tam}$$

1. Accés aleatori : calcules le formula de la pos. de la qual vols l'adreça

2. Accés seqüencial (necesites una manera de parar : A) ÚLTIM ELEMENT DE M

B) L'ELEMENT SEGUENT DEL FINAL C) AMB LA i) \Rightarrow con for's o do while's.

x			
			x
x			

✦ Primera pos. a la que vull accedir (no ha de ser $M[0][0]$)

✦ Últim elem.

✦ Elem. següent

Utilitzem el stride per mourens per M.

* STRIDE : ex: Tenim $M[i][2i+1]$, int $M[4][4]$

	x		
--			x

for (int i=0; i<4 ; i++)

◦ Primera iteració $i = 0$

$M[i][2i+1] \Rightarrow M[0][1]$

◦ Segona iteració $i = 1$

$M[1][3]$

$$\begin{array}{r} M[1][3] \\ - M[0][1] \\ \hline \end{array}$$

★ STRIDE: $\rightarrow M[1][2] \leftarrow$

El calculem:

$$M[0][1] = \&M + 0 \cdot 4 \cdot 4 + 1 \cdot 4 = \&M + 4$$

$$M[1][3] = \&M + 1 \cdot 4 \cdot 4 + 3 \cdot 4 = \&M + 16 + 12 = \&M + 28$$

$$\begin{array}{r} \rightarrow \\ - \\ \hline \end{array} \begin{array}{r} \&M + 28 \\ \&M + 4 \\ \hline 24 \rightarrow \text{STRIDE} \end{array}$$

* Si quiero ir de \rightarrow \leftarrow

la $\$t0, M \quad \# \$t0 = @M[0][0]$

addiu $\$t1, \$t0, 4 \quad \# \$t1 = @M[0][1]$

addiu $\$t1, \$t1, 24 \quad \# \$t1 = @M[1][3]$

SUBROUTINES

1. ESTRUCTURA

- * Crear bloc d'activació (Pila) → si dins la funció crida una altre
- * Desenvolupament funció
- * Desfer bloc d'activació

2. LA PILA

X	→ VARIABLES LOCALS : matrius, vectors i variables petites de les que necessitem l'adreça
X	→ REGISTRES SEGURS : variables que poden ser modificades després de la crida
X	→ \$ra

Els valors es guarden en ordre de declaració SEMPRE i estan ALINEATS.

La pila és sempre múltiple de 4.

* CONSTRUCCIÓ

addiu \$sp, \$sp, -Tamany # reservas espai

sw ... # per guardar el valor actual dels registres segurs

* RESTAURACIÓ

lw... # per recuperar els valors dels registres

addiu \$sp, \$sp, Tamany # restaures l'espai

RENDIMENT

$$CPI_m = \frac{\sum (CPI \cdot n^{\circ} \text{ins})}{\sum (n^{\circ} \text{ins})}$$

$$T_{\text{exe}} = n^{\circ} \text{ins} \cdot CPI_m \text{ segons}$$

$$T_c = \frac{1}{f} \text{ segons} \quad f: \text{freqüència}$$

$$Guany = \frac{T_{\text{exe. ant.}}}{T_{\text{exe. act.}}} = \frac{CPI_{\text{ant}}}{CPI_{\text{act}}}$$

$$P_m = \frac{\sum (P_i \cdot CPI_i \cdot n^{\circ} \text{ins}_i)}{\sum (CPI_i \cdot n^{\circ} \text{ins}_i)} \quad W$$

$$E = P_m \cdot T_{\text{exe}} \cdot J$$

COMA FLOTANT

S	exponent	mantissa
1 bit	8 bits	23 bits

* Es guarden a \$f0, \$f1, ...

* Si $s=0$ num > 0

$s=1$ num < 0

* Exponent → passar 8 bits a decimal -127

* Mantissa → mateix num.

* Normalització → $\pm 1, \text{mantissa} \cdot 2^{\text{exp}}$

* GR5 : si GR5 > 100 ⇒ arrodonim num + 1

< 100 ⇒ es queda igual

= 100 ⇒ parell : res , senar : num + 1

ex: 0.10000000,110 + 20 bits

Si $S=0$ ⇒ signe positiu, altrament negatiu

$$\text{exp} = 0 \times 10 - 127 = 128 - 127 = 1$$

$$\text{mantissa} = \overset{0.5}{1100} \dots + 19 \text{ bits} = 0.5 + 0.25 = 0.75$$

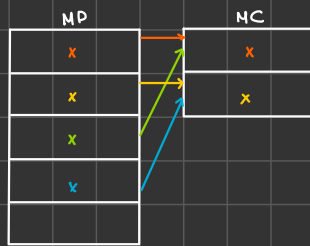
$$\text{NUM} = +1.75 \cdot 2^1 = 3.5$$

MEMORIA CACHE

* TAG / etiqueta → matrícula per identificar la dada

TIPUS D'ASSOCIATIVITAT

CORRESPONDÈNCIA DIRECTA

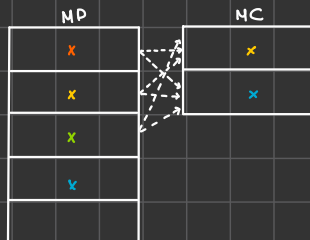


TAG # lineaMC * Byte

* Per saber on es troba una dada a la MC

⇒ $\text{Linea MC} = \text{linea MP} \pmod{\text{Lineas MC}}$

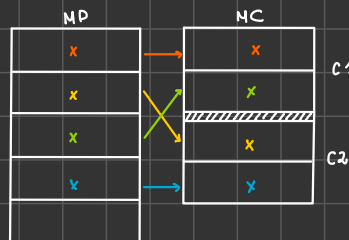
COMPLETAMENT ASSOCIATIVA



* Qualsevol bloc de la MP pot anar a qualsevol bloc de la MC.

TAG # Byte

ASSOCIATIVA DE N-VIAS



* La MC es divideix en conjunts de x-blocs

TAG # conjunt * Byte

ALGORISMES DE REEMPLAÇAMENT

* Aleatori

* FIFO : first input first output, surt el primer que ha entrat

* LRU : last recent used, surt el que porta més temps sense ser usat.

ESCRITURA I LECTURA

		Espectura immediata sense assignació	Espectura immediata amb assignació	Espectura retardada amb assignació
Lectura	Encert	lecMC(byte)	lecMC(byte)	lecMC(byte)
	Fallada	blocMP->MC i lecMC(byte)	blocMP->MC i lecMC(byte)	si DBreemp = 0 blocMP->MC i lecMC(byte) si DBreemp = 1 bloc_reempMC->MP i blocMP->MC i lecMC(byte) i DB = 0
Espectura	Encert	escMC(byte) i escMP(byte)	escMC(byte) i escMP(byte)	escMC(byte) i DB = 1
	Fallada	escMP(byte)	blocMP->MC i escMC(byte) i escMP(byte)	si DBreemp = 0 blocMP->MC i escMC(byte) i DB = 1 si DBreemp = 1 bloc_reempMC->MP i blocMP->MC i escMC(byte) i DB = 1

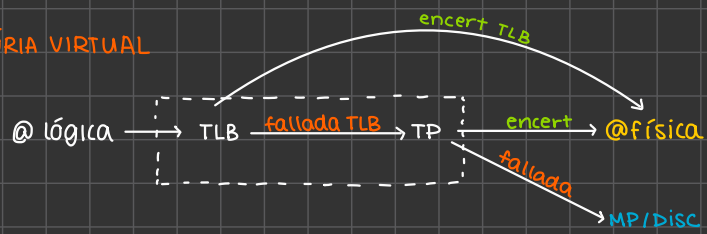
TEMPS

$t_{\text{accés}} = t_h + t_p$

* on el t_p es

tp		Espectura immediata sense assignació	Espectura immediata amb assignació	Espectura retardada amb assignació
Lectura	Encert	0	0	0
	Fallada	tblock + th	tblock + th	si DBreemp = 0 tblock + th si DBreemp = 1 2*tblock + th
Espectura	Encert	0*	0*	0
	Fallada	0*	tblock + th	si DBreemp = 0 tblock + th si DBreemp = 1 2*tblock + th

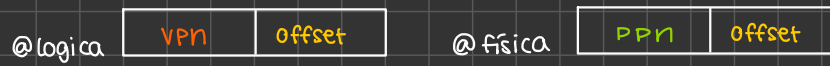
MEMÒRIA VIRTUAL



* TP representa una mini cache del disc

* VPN virtual page number

* PPN phisical page number



* la TP té tantes entrades com VPN ens representen la MP