

## Sistemes Operatius II

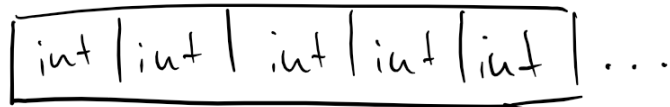
Informe pràctica 1

- A. Explicar, ajudant-se d'un dibuix, la organització de la informació pel cas de les dades de tipus sencers i per què cal accedir a les dades tal com s'indica a les línies 8 i 9 del codi

Aquestes dades, situades a un fitxer de text, són enters. Per poder emmagatzemar-los, hem reservat memòria per un vector d'aquest mateix tipus.

De totes maneres, la capçalera de la funció compara, està formada per dos variables "*const void\**", és a dir, dades de tipus punter *void* o punter genèric. Així, tenim l'opció de poder apuntar a objectes o dades de qualsevol tipus.

En conseqüència, hem de fer un *cast* a cada exercici per especificar el tipus de dada que se l'hi passa. En aquest cas, és tracta d'enters, així que s'accedeix fent *nom\_variable = (int \*) paràmetre1*

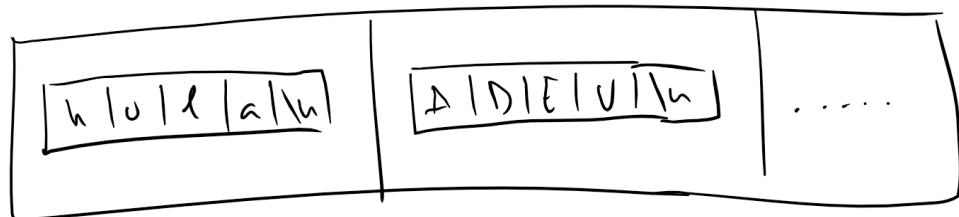


↳ vector[i]

- B. Explicar, ajudant-se d'un dibuix, la organització de la informació pel cas de les cadenes de caràcters i per què cal accedir a les dades tal com s'indica a les línies 10 i 11 del codi 5

La principal diferència aquí es que tractem amb *strings* o cadenes de *chars*. Per poder treballar amb elles, hem implementat un vector de vectors. La capçalera és la mateixa, però requereix un *cast* diferent del anterior.

Treballar amb aquest tipus de vector requereix l'ajuda d'un punter doble, és a dir, *\*\**. Com que és un pas per paràmetre, en fem ús d'un altre punter, *\**. Aleshores queda així: *nom\_variable = \*((char \*\*) paràmetre1*



- C. Suposem que les dades són de tipus sencer, codi 2. Què passa en cas que el casting no es faci de forma correcta? És a dir, què passa si fem un cast a tipus flotant o doble precisió? Explica la teva experiència.

He agafat el codi i he posat *float* en comptes d'*int*. Compilar compila exactament igual. La única diferència és l'aparició de dos *warnings* (un per cada *cast*). Aquests *warnings* t'avisen de que estàs tractant una variable entera com a *float*, i els problemes que en poden derivar.

Per sort, la implementació de la nostra funció "compara" funciona tant per *ints* com per *floats*, així que al executar el programa no sorgeix cap problema, els ordena bé.