



# Pràctica 1: Programes senzills en C

Programació a Baix Nivell — iTIC

Sebastià Vila-Marta

27 de febrer de 2019

## Índex

<b>1 Organització</b>	<b>1</b>
1.1 Material necessari . . . . .	1
1.2 Control de versions . . . . .	1
1.3 Lliurables . . . . .	2
<b>2 Exercicis</b>	<b>2</b>

## 1 Organització

Aquesta sessió s'organitza com una seqüència de problemes de dificultat creixent que van entrenant en l'ús del llenguatge C. La idea és anar-los resolent, implementant i provant un darrera l'altre fins on sigui possible. Us recomanem que en el vostre temps d'estudi els acabeu de resoldre tots. L'objectiu final és anar aconseguint agilitat amb la sintaxi i les eines de treball relacionades amb el llenguatge C. En aquesta pràctica el computador que actuarà de target serà la pròpia estació de treball i el sistema operatiu serà GNU/Linux.

### 1.1 Material necessari

**TASCA PRÈVIA 1.1** Si els resoleu en el vostre computador cal que abans hagueu instal·lat les eines necessàries. En el cas de GNU/Debian i assimilats, cal tenir instal·lats els paquets: `gcc`, `libc6-dev` i `emacs`.

### 1.2 Control de versions

Aquesta pràctica i totes les que segueixen es desenvolupen en equip i amb el suport —obligatori— d'un sistema de control de versions. El sistema a emprar és `subversion`, [CFP11]. Necessàriament cal hostatjar el dipòsit de versions a <https://escriny3.epsem.upc.edu>.

La manera d'organitzar el dipòsit de versions és crear un sol projecte per grup —amb accés als dos membres del grup— i tenir un directori específic per cada pràctica. L'estructura de directoris que n'ha de resultar ha de ser similar a la que s'observa a la figura 1. Recordeu que només se sotmeten a la disciplina del control de versions els fitxers font del projecte i altres fitxers que defineixen el projecte com ara els `Makefile`.

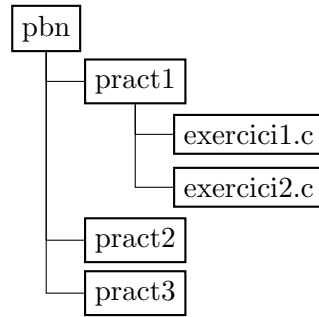


Figura 1: Estructura del dipòsit de versions d'aquest curs.

**TASCA PRÈVIA 1.2** Doneu d'alta un projecte i el corresponent dipòsit de versions a `escriuny3` per a poder gestionar les pràctiques.

### 1.3 Lliurables

Aquesta pràctica no comporta lliurables atès que el domini del llenguatge i les eines són fonamentals per a la resta del curs i resten suficientment avaluades.

## 2 Exercicis

**EXERCICI 2.1** Dissenyeu i implementeu un programa en C99 que escriu pel canal de sortida la frase "No mor qui mor".

**EXERCICI 2.2** Dissenyeu i implementeu un programa en C99 que llegeix pel canal d'entrada un enter  $n$  seguit d'un caràcter  $c$  i escriu pel canal de sortida el caràcter repetit  $n$  vegades.

**EXERCICI 2.3** Dissenyeu i implementeu un programa en C99 que llegeix pel canal d'entrada una frase acabada en el caràcter punt i escriu pel canal de sortida quantes vegades apareix la lletra 'a'.

**EXERCICI 2.4** Dissenyeu i implementeu un programa en C99 que llegeix pel canal d'entrada un byte en format hexadecimal i n'escriu les representacions en base 2, 8 i 10 pel canal de sortida. El programa cal que es digui `converteix`.



**EXERCICI 2.5** Afegiu opcions al programa resultant de l'exercici anterior de manera que `converteix -b` converteixi només a binari, `converteix -o` a octal i `converteix -d` a decimal.

**EXERCICI 2.6** Dissenyeu i implementeu un programa en C99 que llegeix pel canal d'entrada una paraula de 2B en hexadecimal i escriu per la sortida la mateixa paraula després d'haver forçat sengles zeros els bits de més i menys pes. En tot moment la paraula caldrà emmagatzemar-la com a tal i no com una cadena de caràcters o com una taula.

EXERCICI 2.7 Dissenyeu i implementeu un programa en C99 que llegeix pel canal d'entrada una paraula de 2 B en hexadecimal i escriu per la sortida el resultat d'extreure el byte que va dels bits 4 al 11 (el bit de menys pes és el bit 0). En tot moment la paraula i el byte caldrà emmagatzemar-la com a tal i no com una cadena de caràcters o com una taula.

EXERCICI 2.8 Dissenyeu i implementeu un programa en C99 que llegeix pel canal d'entrada una seqüència binària codificada en hexadecimal i acabada en un byte 0 i escriu pel canal de sortida el nombre de bits amb valor 1 d'aquesta seqüència.

EXERCICI 2.9 Considereu que un senyal analògic entre  $-1,0\text{ V}$  i  $1,0\text{ V}$  el mostregeu a  $100,0\text{ Hz}$  i codifiqueu digitalment el resultat sobre 1 B en complement a 2. Una cadena de 100 B, doncs, representa 1 s d'aquest senyal. Assumiu que treballem sempre amb segments d'1 s que els representem com a cadenes hexadecimals.

Dissenyeu i implementeu un programa en C99 que llegeix pel canal d'entrada un segment de senyal i escriu pel canal de sortida un altre senyal de la mateixa natura que l'anterior amb un pols d'amplitud 1 V i 20 ms de durada quan el primer senyal fa un pas per zero.

EXERCICI 2.10 Dissenyeu i implementeu un programa en C99 que llegeix pel canal d'entrada els coeficients d'una equació de segon grau i escriu pel canal de sortida la seva solució.

EXERCICI 2.11 Dissenyeu i implementeu un programa en C99 que llegeix pel canal d'entrada un preu en euros i escriu pel canal de sortida el desgloss mínim en moneda que correspon al preu.

## Referències

[CFP11] Ben Collins-Sussman, Brian W. Fitzpatrick i C. Michael Pilato. *Version Control with Subversion*. Ang. Vers. 1.7. 2011. URL: <http://svnbook.red-bean.com/en/1.7/svn-book.pdf> (cons. 20-02-2019).