Pràctica 1: Programes senzills en C

Programació a Baix Nivell — Enginyeria de Sistemes TIC

Sebastià Vila-Marta

19 de febrer de 2025

Índex

1 Organització

Aquesta sessió s'organitza com una seqüencia 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 installats 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://escriny.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`.

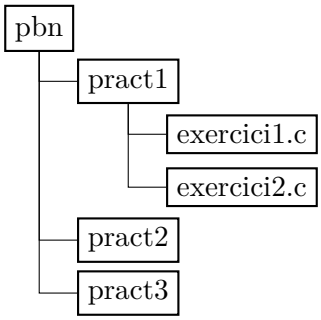

Figura 1: Estructura del dipòsit de versions d'aquest curs.

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

1.3 Mètode de treball

En aquesta pràctica i totes les que segueixen es treballarà dins l'equip seguint el mètode que s'explica a continuació de forma obligada. El mètode té per objectiu obtenir un bon rendiment i, a la vegada, una bona qualitat final del programari escrit. El mètode de treball cal aplicar-lo amb el suport del sistema de control de versions.

El mètode en qüestió segueix les següents fases:

1. Les persones de l'equip estudien l'enunciat pel seu compte.

2. Fase de planificació.

   L'equip es reuneix i decideix com repartir la feina de cada pràctica. Hi ha moltes maneres de fer-ho: per mòduls, per funcions de cada mòdul, etc. Forma part de l'estratègia de cada equip com es faci. També decideix el calendari de treball per a la pràctica en qüestió. Noteu que la feina inclou la fase d'integració i la feina de lliurament, que també cal assignar-les explícitament. Pel que fa a la fase d'integració, es pot assignar a una part de l'equip o abordar-la col·lectivament.

   Cal deixar nota escrita de la feina que assumeix cada persona de l'equip en aquesta reunió així com del calendari pactat.

3. Fase de desenvolupament.

   Cada persona aborda de manera individual les tasques que són de la seva responsabilitat en el termini que hagi fixat l'equip. Durant aquesta etapa, cada persona actualitza la seva part en el dipòsit de versions de l'equip, sempre amb el seu usuari i mai amb l'usuari d'una altra persona. Els commentaris dels «commit» han de ser acurats i descriptius i han d'incloure explícitament el tag `#devel`.

4. Fase de verificació.

   Cada persona de l'equip revisa explícitament la feina feta pels altres. Com a resultat pot fer-hi modificacions i millores o corregir-ne errors.

   La revisió es fa especialment via la lectura analítica del codi escrit per la resta de persones de l'equip per descobrir aspectes dubtosos o de qualitat millorable. Es pot comentar amb els autors els aspectes que siguin necessaris.

Com a resultat de la revisió es poden produir noves versions dels fonts que cadrà afegir al sistema de versions sota la identitat del responsable de la revisió sempre. De manera sistemàtica, s'afegira una nova versió (`commit`) per cada qüestió revisada i, en el missatge de la versió s'hi farà constar el tag `#review`.

5. Fase d'integració i test.

   L'equip es reuneix per integrar tot el que han escrit els diferents membres en el producte final. El producte final es prova extensivament de manera sistemàtica. Els errors que es detectin, si n'hi ha, es corregeixen. Cada correcció genera una nova versió el comentari de la qual es marca amb el tag `#bug`. Generalment ha d'haver-hi pocs errors. Un excés d'errors significa que la feina de les fases anteriors no és prou bona.

6. Fase de lliurament.

   La persona que tingui assignada la feina prepara el paquet amb el contingut que es demana i el format pactat i el lliura.

Els fitxers font que s'escriguin començaran sempre per una petita capçalera en que s'hi anotarà qui és el desenvolupador i qui i quan l'ha revisat seguint aquest model:

```
/*
Developer: Sebas Vila (2-feb-2020)
Revised: Paco del Àguila (7-feb-2020)
*/
```

L'implementador escriurà la seva part de la capçalera i el revisor l'afegira en el moment de fer la revisió tant si se'n deriven canvis com si no.

1.4 Lliurament

El lliurament de la pràctica —i de la resta— consisteix a enviar a Atenea un fitxer empaquetat (`tar`) i comprimit (`gzip`). El nom del fitxer que lliureu ha de ser exactament `pract-P-equip-N.tar.gz` on `P` és el número de pràctica —en aquest cas `1`— i `N` és el número del vostre equip. El contingut del fitxer ha de ser exactament el següent:

1. Un fitxer de text de nom `README` que conté informació sobre:
   - Qui forma l'equip.
   - L'URL del dipòsit de versions amb que treballen.
   - L'esquema de repartició de tasques i el calendari.
   - El *log* de subversion corresponent a totes les fases del projecte.

   La figura 2 mostra un exemple de fitxer `README`.

2. Els fitxers font propis de la pràctica.

3. El `Makefile` corresponent una vegada s'hagi introduit l'eina.

El lliurament no ha de contenir mai:

1. Fitxers objecte o executables.

2. Enunciats o altres documents.

```
Pràctica 1 PBN
==============

Dades generals
--------------

- Sebastià Vila
- Paco del Àguila

https://escriny.epsem.upc.edu/projects/sebastia.vila/repository/p1

Planificació
------------

Sebas: Assumeix les feines de ...

Paco: Assumeix les feines de ...

Log de versionat
----------------

------------------------------------------------------------------------
r557 | sebastia.vila | 2020-02-20 09:41:11 +0100 (dj, 20 feb 2020) | 1 line

Canvi escriny3 a escriny.
------------------------------------------------------------------------
r550 | sebastia.vila | 2019-06-12 21:57:34 +0200 (dc, 12 jun 2019) | 1 line

Purga de proves diverses.
------------------------------------------------------------------------
r539 | sebastia.vila | 2019-02-27 14:21:16 +0100 (dc, 27 feb 2019) | 1 line

Afegida referencia.
------------------------------------------------------------------------
r538 | sebastia.vila | 2019-02-27 13:17:17 +0100 (dc, 27 feb 2019) | 3 lines

Afegit apartat sobre svn.
------------------------------------------------------------------------
r365 | jordi.bonet | 2017-02-24 13:03:33 +0100 (dv, 24 feb 2017) | 1 line

Actualitzacions menors.
------------------------------------------------------------------------
r364 | jordi.bonet | 2017-02-24 00:09:47 +0100 (dv, 24 feb 2017) | 1 line

Afegides solucions alternatives.
------------------------------------------------------------------------
r360 | jordi.bonet | 2017-02-23 17:26:08 +0100 (dj, 23 feb 2017) | 1 line

Typos i correccions llengua.
------------------------------------------------------------------------
r357 | jordi.bonet | 2017-02-20 02:28:11 +0100 (dl, 20 feb 2017) | 1 line

Correccions gramaticals.
------------------------------------------------------------------------
r352 | jordi.bonet | 2017-02-18 19:53:29 +0100 (ds, 18 feb 2017) | 1 line

Aportades solucions als exercicis proposats.
------------------------------------------------------------------------
r350 | jordi.bonet | 2017-02-15 21:39:30 +0100 (dc, 15 feb 2017) | 1 line

Millora de l'enunciat.
------------------------------------------------------------------------
r322 | sebastia.vila | 2016-04-05 14:04:54 +0200 (dt, 05 abr 2016) | 1 line

Reorganitzada l'estructura de directoris i makefiles.
------------------------------------------------------------------------
```

Figura 2: Exemple de fitxer `README`

3. Backups de l'editor.

4. Proves o altres elements que no són part del producte final.

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 2 B 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 $-1{,}0$ V i $1{,}0$ V el mostregeu a $100{,}0$ 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 desglòs mínim en moneda que correspon al preu.

Referències

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