

PAC3: Control de Versions i Documentació

Josep Turón i Viñas

Desembre del 2015

1 Introducció

Aquest document, realitzat amb \LaTeX , explica de forma detallada com s'ha portat a terme la resolució de la PAC 3: Control de Versions i Documentació, de l'assignatura optativa Desenvolupament de Programari, corresponent al Màster Universitari en Programari Lliure.

En la primera part, s'explicarà com s'ha creat un dipòsit utilitzant Git amb els fitxers de filtres utilitzats en les pràctiques anteriors.

En la segona part, s'explicarà com s'ha realitzat la documentació del codi dels filtres, utilitzant Doxygen, i finalment com s'ha redactat l'explicació de la PAC amb \LaTeX .

Finalment es presentaran unes breus Conclusions sobre el desenvolupament de la pràctica.

Aquest document s'entregarà en format `tex`, `dvi` (propi del visor de textos `xdvi`) i `pdf`, juntament amb el fitxer de log de la compilació del fitxer `tex`. Tots aquests s'adjuntaran dins el fitxer `PAC3.tar.gz` juntament amb tots els altres fitxers generats durant el desenvolupament de la PAC.

Un cop descomprimit el fitxer `PAC3.tar.gz` es generarà la següent estructura de directoris:

- `/PAC3`: conté el fitxer `configdoxy` de configuració de Doxygen.
- `/PAC3/libsrc`: conté els fitxers amb el codi de les funcions `filtres.c` i la capçalera `filtres.h`.
- `/PAC3/doc`: conté la documentació del codi dels filtres generada amb Doxygen i exportada en format HTML i \LaTeX .
- `/PAC3/descripcio_proces`: conté la documentació generada amb \LaTeX amb l'explicació del desenvolupament de la pràctica, en formats `tex`, `dvi` i `pdf` (aquest fitxer).
- `/PAC3/.git`: conté el dipòsit de Git (directori ocult).

2 Control de Versions

Per començar hem instal·lat les eines **git** i **gitk** (aquesta ens permet visualitzar el contingut del dipòsit i els *commit* realitzats de forma gràfica), executant les comandes:

```
sudo apt-get install git
```

```
sudo apt-get install gitk
```

A continuació configurem l'usuari, el seu correu electrònic, i un editor de textos per defecte. Així mateix podríem definir altres eines, que no necessitem en aquesta pràctica:

```
git config user.name Josep
```

```
git config user.mail jturonv@uoc.edu
```

```
git config core.editor vim
```

Ara ja podem crear el dipòsit fent:

```
git init
```

I afegir els dos fitxers que volem a l'*staging area*:

```
git add libsrc/filtres.c
```

```
git add libsrc/filtres.h
```

o simplement pujant el directori:

```
git add libsrc
```

Amb la comanda

```
git status
```

veiem que tenim els dos nous fitxers pendents que encara no s'han posat al dipòsit. Fem un *commit* i els passem de l'*staging area* al dipòsit:

```
git commit -m "primera pujada"
```

L'opció -m ens permet afegir un comentari que ajudarà a la persona que revisi el dipòsit a entendre ràpidament quins canvis s'han fet en aquest *commit*. Ara fent altre cop `git status` podem veure com els fitxers ja han passat al dipòsit.

Executant `gitk` podem comprovar de forma gràfica tots els *commit* fets al dipòsit, els canvis realitzats, l'estructura d'arbre del projecte, etc. (veure figura 1).

Amb això, hem configurat un dipòsit en local, que és com treballa Git. Per tal de que estigui disponible a altres usuaris hem creat també un dipòsit remot. Hem accedit a GitHub (<https://github.com>) i hem registrat l'usuari `jturonv@uoc.edu`. Després afegim l'origen remot:

```
git remote add origin http://github.com/jturonv/PAC3.git
```

i per últim pugem el dipòsit local a GitHub (amb l'últim *commit* que haguem fet en local):

```
git push origin master
```

 (demana usuari i contrasenya de l'usuari creat a GitHub)

Ara podríem clonar aquest dipòsit remot en local a qualsevol ordinador. Si fem:

```
git clone http://github.com/jturonv/PAC3.git
```

Crearem una carpeta anomenada PAC3 dins el directori PAC3 on estem treballant que contindrà la mateixa estructura de carpetes i contingut que tenim en el dipòsit local (no s'adjunta per tant en el fitxer `PAC3.tar.gz` de la pràctica).

Per acabar, el control d'accés d'usuaris al dipòsit no forma part pròpiament de Git, però accedint a la web de GitHub es poden configurar col·laboradors per al projecte, simplement escollint-los d'entre tots els usuaris registrats a GitHub.

La imatge següent il·lustra la utilització de la eina **gitk** que mostra el contingut del dipòsit de Git amb l'estructura de diverses proves de *commit* realitzades i els canvis fets a cadascuna.

l'explicació completa del filtre. La documentació de cada filtre quedarà enllaçat amb el codi font, podent accedir ràpidament a un o altre.

Fet això, només queda generar la documentació en HTML com havíem configurat i afegir-la al dipòsit. Per generar-la només cal executar la comanda:

```
doxygen configdoxy
```

I es crearà el directori doc amb dos directoris, un anomenat html amb la documentació en aquest format i un altre, latex, per poder-la crear utilitzant aquesta eina. Si accedim al directori html, només cal obrir el fitxer index.html amb un navegador web per a consultar la documentació generada.

Acabem afegint al dipòsit aquest directori doc amb la documentació, i actualitzant els fitxers de codi ja existents amb les últimes versions:

```
git add libsrc
```

```
git add doc
```

Afegim els dos directoris i tots els fitxers que contenen. Amb git status veiem els fitxers nous que s'afegiran al fer el *commit*:

```
git commit -m "segon commit, amb documentació"
```

I ja només queda actualitzar també el dipòsit remot a GitHub, fent un *push* del local:

```
git push origin master
```

La segona part consisteix en documentar en un fitxer mitjançant el processador de textos **LaTeX** el treball desenvolupat en aquesta pràctica (aquest fitxer). Això ho farem amb text pla sense format, utilitzant etiquetes pròpies de LaTeX que el programa compilarà per a obtenir el text en el format escollit (en el nostre cas hem escollit fer-ho com un article).

Comencem a instal·lar LaTeX amb el nombre màxim d'opcions:

```
sudo apt-get install texlive-full
```

Després procedim a editar un fitxer en blanc anomenat PAC3.tex ubicat dins el directori descripcio_proces. Configurem algunes opcions com la llengua (català), tipus de codificació de caràcters, tipus de document (article) i l'opció per poder utilitzar els accents i caràcters especials catalans normalment. Definim el títol, autor i data, els diferents apartats que el componen, i comencem a redactar el document.

Un cop finalitzat el compilem fent:

```
latex PAC3.tex
```

generant així un fitxer dvi, que podem visualitzar en format gràfic amb l'eina xdvi:

```
xdvi PAC3.dvi
```

Com que també volem tenir el fitxer en format pdf, el creem executant:

```
pdflatex PAC3.tex
```

I per acabar, només resta pujar el directori descripcio_proces amb aquesta documentació als dipòsits local i remot. Fem:

```
git add descripcio_proces
```

```
git commit -m "final de projecte"
```

```
git push origin master
```

Tot i que no l'hem utilitzat per a crear el fitxer, hem provat també l'eina **LyX**, un editor de textos WYSIWYG (What-You-See-Is-What-You-Get) que permet visualitzar el codi editat amb format mentre es va desenvolupant sense necessitat de compilació final (com sempre, cal instal·lar-lo abans fent `sudo apt-get install lyx`).

4 Conclusions

En la realització d'aquesta pràctica hem assolit els següents objectius:

A la primera part hem après a utilitzar un sistema de control de versions i a diferenciar-ne els dipòsits locals i remots. En el nostre cas hem utilitzat Git, el qual treballa amb una còpia local, en la qual els fitxers afegits van a parar a l'*staging area* que són pujats al dipòsit (local) fent un *commit*. Podem també crear un dipòsit remot, per exemple a GitHub, disponible perquè altres usuaris el puguin clonar com a còpia local.

A la segona part, hem après a utilitzar Doxygen, un sistema de creació de documentació integrada amb el codi font i exportable a diversos formats, entre ells HTML, per tal de generar automàticament una documentació de tot el codi desenvolupat amb referències creuades entre tots els fitxers del projecte.

Finalment, a la tercera part hem après a redactar documents amb el processador de textos LaTeX i a diferenciar-ne el contingut del disseny final, ocupant-nos només de redactar aquest contingut i deixant que l'eina compili els codis introduïts amb el text per a crear-ne automàticament el disseny del document, segons el tipus escollit i indicat.