

## Taller Sistemas Operativos 06 a Octubre 2017

### 6.1 - programas C edición<sup>1</sup> (y compilación)<sup>2</sup>

Seguimos editando (y compilando, y ejecutando) programas.

**vi**-ene bien recordar que **x dw dd** borran carácter, palabra y línea, respectivamente; que **p P** traen lo borrado, antes o después del cursor, que **xp ddp** intercambian carácter y línea, y que **u** deshace un cambio. (y **vimtutor**)

Si se comete algún error habrá que repetir las líneas 40, 50 (y quizá 60).

Procedimientos, funciones de **c**, ordenando líneas ...

```
cd ; cd tso/01/progs
cat fi02.c
vi fi02.c           # 40
make fi02          # 50
./fi02            # 60
```

Máximo común divisor,

```
cd ; cd tso/01/progs
cat fi03.c
vi fi03.c
make fi03
./fi03
```

### 6.2 - sed<sup>3</sup> edición no interactiva

**sed** recorre todas las líneas de entrada.

las ordenes/comandos de **sed** constan de:

*parte dónde*      *acción*      *parámetros*

a) **parte dónde**: nula, número, 2 números, expresión regular, 2 expr.reg.

parte dónde negada: **!** .

Una acción: **borrar**, indicado por **d** (*delete*).

A veces se puede prescindir de las comillas. Y ... se pueden dejar.

```
cd ; cd tso/01/palabras
sed -e '3d' dias           ; sed -e '3d' meses
sed -e '3!d' dias         ; sed -e '3!d' meses
sed -e '3,5d' dias        ; sed -e '3,5d' meses
sed -e '3,5!d' dias       ; sed -e '/es/d' dias
sed -e '/es/!d' dias      ; grep 'es' dias
sed -e '/r/,/u/d' dias    ; sed -e '/r/,/u/!d' meses

sed '3,9d' dias           ; sed 3,9d dias
```

b) Puede haber varias ordenes de **sed** en línea de comandos o *en un fichero*.

```
sed -e '2,4d' -e '7,9d' meses

echo '2,4d' > p1.sed
echo '7,9d' >> p1.sed
sed -f p1.sed dias       ; sed -f p1.sed meses
```

<sup>1</sup>apuntes SSAA, cap. 8, pag. 48-51

<sup>2</sup>apuntes SSAA, cap. 23, pag. 269

<sup>3</sup>apuntes SSAA, cap. 12, pag. 81-86

c) El primer `-e` se puede omitir.

```
sed -e '5,$d' dias ; sed -e '5,$d' meses
sed '5,$d' dias ; sed '5,$d' meses
```

```
cat dias ; cat meses
```

`$` en la *parte dónde* de `sed` significa: última línea.

d) El **fichero de entrada no se modifica**. (Si no indicamos lo contrario).  
Podemos guardar el resultado mediante redirección.

```
sed -e '6,7d' dias > trabajo
```

e) Otra acción/verbo: **substituir**, indicado por `s` .

```
sed -e 's/es/ES/' dias
sed -e 's/r/R/' meses
```

(continuará)

### 6.3 - iteración `bash` `escript`, `for` <sup>4</sup>

Creas un fichero de nombre `fvocals`

```
cd ; cd tso/01/mio
echo for i in a e i o u > fvocals
echo do >> fvocals
echo echo '$i' >> fvocals
echo done >> fvocals
bash fvocals
```

Cambias (400) el orden de las vocales en la primera línea para que la salida sea `a i u e o` , una letra en cada línea.

```
vi fvocals # 400
bash fvocals
```

Escribes el `for` en la línea de comandos. (Un poco *friki* ;-)

```
for i in u o i e a ; do echo $i ; done
```

Cambias en `fvocals` la línea de `echo $i` por las dos líneas siguientes:

```
echo " $i "
grep $i ../palabras/dias | head -2
```

y ejecutas con `bash fvocals` .

---

<sup>4</sup>apuntes SSAA, cap. 22, pag. 236-237