

**9.1 - awk**<sup>1</sup>

`awk` tiene varias funciones para trabajar con *tiras de caracteres* o *strigs*. A una de las funciones se asocia un *operador invisible*

```
cd ; cd tso/01/awks
awk '{print substr($1,1,4)}'    dias ; awk '{print substr($1,3,4)}' dias
awk '{print substr($1,3)}'      dias

awk '{print length($1), $1}'    dias

awk '{print index($1,"e"),$1}'  dias ; awk '{print index($1,"r"),$1}' meses

awk '{print $4, $3, $2, $1
      print $4 $3 $2 $1}'
uno dos tres cuatro
a b c d
^D                                # control-D !

awk '{n = split($1, v, "i")
      print n, v[1], v[2], v[3], v[4]}'  ../palabras/largas
```

**9.2 - redirección**<sup>2</sup>

Aunque se usen menos vamos a ver << y tee .

```
cd; cd tso/01/mio
cat >f1 <<FIN
  bla
  ble bli blo
  blu
FIN
cat f1

pr -t -m f1 f1 f1
pr -t -m f1 f1 f1 | tee f3
cat f3

grep 'e.*e' ../palabras/dias | nl
grep 'e.*e' ../palabras/dias | tee f5 | nl
cat f5

date; who | head -5 > cuandoyquien          # 540
cat cuandoyquien
( date; who | head -5 ) > cuandoyquien      # 560
cat cuandoyquien
```

---

<sup>1</sup>apuntes SSAA, cap. 17, pag. 150-151, 161

<sup>2</sup>apuntes SSAA, cap. 11, pag. 65-71

**9.3 - bash**              **test** [ <sup>3</sup>

El mismo comando admite dos *sintáxis* (escrituras):

**test condición(es)**        y        **[ condición(es) ]**

```

cd; cd tso/01/progs/bashes
rm ? ??

test -r a ; echo $?
date > a ; test -r a ; echo $?
test 5 -eq 5 ; echo $?
test 7 -gt 5 ; echo $?
test 07 -ge 5 ; echo $?      # 660
test 07 = 5 ; echo $?       # 670
test -f a ; echo $?

if [ -r a ] ; then echo legible ; else echo no legible ; fi
if [ -d .. -a -r .. ]
then echo dir legible
else echo dir no legible
fi

cal >b
[ -r a ] && cp b c           # 740
cat c

```

---

<sup>3</sup>apuntes SSAA, cap. 21, pag. 242-244