

BesturingsSystemen

Les Uno

Commando's deel 1

Touch, comm & diff

touch – the easiest way to create new, empty files. It is also used to change the timestamps

comm – compare two sorted files line by line **diff** – find differences between two files

```
touch [-c] [-r ref] [-t [YY]MMDDhhmm[.ss]]
    -c    enkel indien bestand reeds bestaat
    -r    transfereert tijdsinformatie van ref
    -t    willekeurige tijdsinformatie

comm [-1] [-2] [-3]
    -1    geen lijnen enkel aanwezig in bestand1
    -2    geen lijnen enkel aanwezig in bestand2
    -3    geen lijnen aanwezig in beide bestanden

diff [-i] [-w] [-B] [-c|u[#]] [-y]
    -i    niet gevoelig voor hoofdletters
    -w    negeert witruimte in niet-lege lijnen
    -B    negeert volledig lege lijnen
    -c[#] extra lijnen context rond verschillen
    -u[#]  alternatieve representatie voor -c
    -y [--suppress-common-lines] uitvoer in 2 kolommen
```

ls, cp, mv & rm

ls – list directory contents **cp** – copy files and directories **mv** – move (rename) files **rm** – remove files or directories

```
ls [-l] [-a|A] [-d] [-t|S] [-r] [-R] [-1] {target}
    -l    detailinformatie
    -a    ook bestanden startend met .
    -A    idem, behalve . en ..
    -d    mappen zelf, niet hun inhoud
    -t    gesorteerd, meest recente eerst
    -S    gesorteerd op size
    -r    sorteervolgorde omkeren
    -R    recursief
    -1    niet in kolommen (1 lijn per bestand)
```

```
cp/mv/rm [-i|f] [-u] [-v] [-p] [-r|a] target
-i      interactief (bij overschrijven target)
-f      negeert/omzeilt waarschuwingen
-u      niet uitgevoerd indien doel recenter
-v      verbose modus
-p      laat klassieke attributen intact
-r      recursief
-a      recursief + laat alle attributen intact
```

Shuf, paste, wc

shuf – generate random permutations **paste** – merge lines of files

```
shuf [-r] [-n ...] [-e {...}]
-e      treat each ARG as an input line
-n      --head-count=COUNT output at most COUNT lines
-r      random
paste [-d ...] [-s]
-d      stelt delimiter(s) in
-s      elk bestand op één lijn
wc [-c] [-w] [-l] [-L]
-c      telt bytes
-w      telt woorden
-l      telt lijnen
-L      toont lengte van de langste lijn
```

Voorbeelden deel 1

```
//Show files met details
ls -l {0..9}

//Verwijder files met naam 0,1,2,3,...
rm -f {0..9}

//kopieer file1 naar file2
cp file1 file2

//Maakt lege files aan met naam 0,1,2,3,...
touch {0..9}

//Update bestaande files (-c) met naam.. en gebruik als tijdsref. (-r) file qq
touch -cr qq {0..9}

//Update bestaande files met naam.. en gebruik als datum (-t) 31 jan 12:34
touch -ct 01311234 {0..9}

//Kies 20 (-n) random (-r) letters uit a..j (-e)
shuf -rn 20 -e {a..j}

//Kies 17... Sorteert en save in x
shuf -rn 17 -e {a..j} | sort > x
```

```

//Inhoud van 2 bestanden naast
paste x y

//Inhoud van bestanden xyz met delimiters (-d) uitvoer bv 1;2:3.4;5:6. enz. (;: h
erhalen)
paste -d \;:. x y z

//Alle kolommen op 1 rij
paste -s -d \;:. x y z

//vergelijk 2 gesorteerde bestanden (Linkerkolom enkel in bestand 1, midden enkel i
n bestand 2, rechts in beide bestanden)
comm x y
//vergelijk inhoud van x en y en geef resultaat van wat enkel in bestand 2 te vinde
n is (-13 negeert kolom 1 en 3)
comm -13 x y

//Geef verschillen tussen 2 ongesorteerde bestanden
diff file1 file2

//Geef verschillen case insensitive (-i), geen whitespaces (-w), geen newlines (-B)
uitvoer naar wc en tel lijnen (-l) ->tel aantal verschillende lijnen
diff -iwB file1 file2 | wc -l

//Vergelijk in blokken van 2 (-c 2)
diff -c2 file1 file2

//vergelijk naast mekaar, enkel de verschillen tonen
diff -wBy --suppress-common-lines file1 file2

```

Commando's deel 2

sha224sum

sha224sum - compute and check SHA224 message digest

```

❏ sha224sum [OPTION] [FILE]
    -b      read in binary mode
    -c      read SHA224 sums from the FILEs and check them
    --tag   create a BSD-style checksum
    -t      read in text mode (default)

```

split, join

split - split a file into pieces **join** - join lines of two files on a common field

```

❏ split [-l #] [-b|C #] [-a#] [-d] [prefix]
    -l # maximaal # lijnen in fracties
    -b # maximaal # bytes in fracties
    -C # idem, maar enkel volledige lijnen

```

- a bepaalt lengte suffixes fractienamen
- d numerieke suffixes

```
join [-t ...] [-1 ...] [-2 ...] [-o ...] [-a|v 1|2]
-t bepaalt velddelimiter
-1 en -2 stelt ON predicaat in
-o bepaalt uitvoerveld(en)
-a simuleert left/right outer join
-v simuleert outer join, beperkt uitvoer
  tot lijnen zonder overeenkomst
```

locate, find

locate – list files in databases that match a pattern

```
❏ locate [-i] [-b] [-r]
-i niet gevoelig voor hoofdletters
-b zoekt in bestandsnaam, niet in pad
-r reguliere expressie i.p.v. patroon

find [-H] [-L] [-P] [path...] [expression]
-P volg nooit symbolische links
-L volg symbolische links
-H volg nooit symbolische links tenzij....
```

xargs

xargs – build and execute command lines from standard input

```
❏ xargs [-I ...] [-p|t] [-n#] ...
-n# consumeert maximaal # argumenten
-I ... stelt placeholder voor argument in
-i equivalent met -I {}
-p interactieveodus
-t toont ook uitgevoerde opdrachten
```

Voorbeelden deel 2

```
❏ //Bereken checksum op basis van inhoud voor elke file in current map, save result i
n bestand t
sha224sum ?? > t

//Check files in current map checksum met checksums in t
sha224sum -c t

//Split laat toe om files te splitsen op lijnaantal / bytegrootte
//Om bestand te gebruiken als invoer nooit 'cat x | opdr' maar wel 'opdr < x'
// x is invoer bestand, y is uitvoerbestandprefix, splitsen op 1 lijn (-l 1)
split -l 1 x y

// x is invoer bestand, uitvoerbestandprefix is een nummer, splitsen op 1 lijn (-l
```

```

1)
split -l 1 x '

//split bestand obv bytegrootte met K 1024bytes -C 10K, numerieke prefix (-d)
split -d -C 10K x

//Vind een bestand met TODO in de naam (-b), zonder -b wordt gezocht in volledige p
adnaam, maar werkt met snapshot 1x per dag
locate -b TODO

//zoeken met met regexp
locate -br TODO$

//zoek naar bestanden die aan alle voorwaarden worden voldaan (AND)
find /usr/share -name "*TODO*"

//naam bevat TODO, groter dan 10kb, laatst gewijzigd < 200dagen
find /usr/share -name "*TODO*" -size +10k -mtime -200

//groter dan 10kb OF laatst gewijzigd < 200dagen
find /usr/share -name "*TODO*" \( -size +10k -o -mtime -200 \)

//print uit size in 8 getallen, pad en newline
find /usr/share -name "*TODO*" -size +10k -mtime -200 -printf "%8s %p\n"

//count words in them files
wc $(find /usr/share -name "*TODO*" -size +10k -mtime -200)

//groepeer per 3 groepjes tel per element en een totaal per groepjes
find /usr/share -name "*TODO*" -size +10k -mtime -200 | xargs -n3 wc

//vraag of groepje moet uitgevoerd worden of niet y/n
find /usr/share -name "*TODO*" -size +10k -mtime -200 | xargs -pn3 wc

//Moet ik wc doen of niet? met {} de huidige filename, \; om de opdracht af te sluit
en
find /usr/share -name "*TODO*" -size +10k -mtime -200 -ok wc {} \;

//Alle opdrachten een voor een uitvoeren, zoeken counten, zoeken counten,..
find /usr/share -name "*TODO*" -size +10k -mtime -200 -exec wc {} \;

//Alle opdrachten uitvoeren, eerst zoeken dan count op alles
find /usr/share -name "*TODO*" -size +10k -mtime -200 -exec wc {} \+

*The moreau way* //Efficiente manier
find /usr/share -name "*TODO*" -size +10k -mtime -200 -printf "wc %p\n" | bash

```

Commando's deel 3

cat, tac & rev

cat – concatenate files and print on the standard output **tac** – concatenate and print files in reverse **rev** – reverse lines characterwise

```
❏ cat/rev [-n|b] [-s] [-v] [-T] [-E]
    -n    ook blanco lijnen genummerd
    -b    blanco lijnen niet genummerd
    -s    opeenvolgende blanco lijnen als één
    -v    toont alle controletekens als ^
    -T    tabs als ^I getoond
    -E    $ teken na elke lijn

    tac [-[r]s ...]
        -s    alternatieve scheidingsstring
        -rs    reguliere expressie als scheiding
```

sort

sort – sort lines of text files

```
❏ sort [-t...] [-k...] [-f] [-n] [-r] [-o...] [-C] ...
    -t    stelt velddelimiter in
    -k    stelt sorteersleutel(s) in
    -f    mapt kleine letters op hoofdletters
    -n    numeriek i.p.v. alfabetisch
    -r    sorteervolgorde omkeren
    -o    uitvoerbestand (mag = invoerbestand)
    -c|C  controleert of invoer gesorteerd is, al
           dan niet met vermelding eerste fout
```

wc

wc – print newline, word, and byte counts for each file

```
❏ wc [-c] [-w] [-l] [-L] ...
    -c    telt bytes
    -w    telt woorden
    -l    telt lijnen
    -L    toont lengte van de langste lijn
```

Voorbeelden deel 3

```
❏ #Twee bestanden joinen
//Eerst sorteren
//sorteer met delimiter ';' (-t \;) vanaf het derde veld -k3 tot het einde van de 1
ijn uit het bestand w
sort -t \; -k3 w

//sorteer vanaf het derde veld -k3 tot het 3e veld
sort -t \; -k3,3 w
```

```

// -o w w input w output w !uitzonderlijk, r reverse sorteren, n lexicograaf sortere
n
sort -t \; -k3,3nr -o w w

//joinen
//join obv kolom 3 1e bestand en kolom 2 2e bestand -1 3 -2 2, bestand w en c, -o
geef volgende kolommen
jon -t \; -1 3 -2 2 w c -o 2.4,2.1,1.2,1.4,1.1

//join obv kolom 3 1e bestand en kolom 2 2e bestand -1 3 -2 2, bestand w en c, -v e
nkel degene die niet gevonden worden, -a alle
jon -t \; -1 3 -2 2 w c -v 2

#Maak matrix en invertteer
//Geef 36 random getallen tussen 0-9
shuf -rn36 -i 0-9

//Groepeer in groepen van 6 horizontaal (echo werkt als default)
shuf -rn36 -i 0-9 | xargs -n6 echo

//Save in bestand
shuf -rn36 -i 0-9 | xargs -n6 > x

//Splits in groepjes van 6 verticaal
shuf -rn36 -i 0-9 | xargs -n1 | split -l 6

//tel woorden
wc - w < winners.csv

//tel woorden per bestand en totaal
wc file1 file2

// toon lengte van langste lijn
wc -L file1 file2

```

Opmerkingen

- verschil change en modify: modify is inhoud veranderd (checksum verandert), change attributen veranderen checksum zelfde
- Gebruik nooit zelfde bestand als invoer en uitvoer bv sort < w > w mag niet