

Data-Entry documentatie

GEBRUIKERSGIDS SYSTEEM-COMMANDO'S

Een wegwijzer bij het gebruik van een aantal systeem-commando's, ten behoeve van hen die werken met behulp van het "Data-Entry"-systeem op de PDP11/34 - computer van de Stichting Mathematisch Centrum.

Samengesteld door :

H.Noot
G.J.F.Vinkesteijn
A.C.IJsselstein

Data-Entry documentatie

GEBRUIKERSGIDS SYSTEEM-COMMANDO'S

Een wegwijzer bij het gebruik van een aantal systeem-commando's, ten behoeve van hen die werken met behulp van het "Data-Entry"-systeem op de PDP11/34 - computer van de Stichting Mathematisch Centrum.

Samengesteld door :

H.Noot
G.J.F.Vinkesteijn
A.C.IJsselstein

INHOUD:

HOOFDSTUK 0	INLEIDING	81/01/29
HOOFDSTUK 1	BEKIJKEN EN VERGELIJKEN VAN FILES	81/01/29
	cat	81/01/29
	cmp	81/01/29
	difff	81/01/29
	grep	81/01/29
	more	81/01/29
	spell	81/01/29
	wc	81/01/29
HOOFDSTUK 2	PRINT-PROGRAMMA'S	81/01/29
	hqp	81/01/29
	lpr	81/01/29
	pr	81/01/29
	print	81/01/29
	pt	81/01/29
	qupr	81/01/29
	rdcas	81/01/29
	wrcas	81/01/29
HOOFDSTUK 3	GEGEVENSINVOER-PROGRAMMA'S	81/01/29
	bibed	- - - -
	concept	81/01/29
	ded	81/01/29
	editor	81/01/29
HOOFDSTUK 4	HET FILE-SYSTEEM	81/01/29
	cd	81/01/29
	chdir	81/01/29
	chmod	81/01/29
	cp	81/01/29
	ls	81/01/29
	mkdir	81/01/29
	mv	81/01/29
	pwd	81/01/29
	rm	81/01/29
	rmdir	81/01/29
HOOFDSTUK 5	DIVERSEN	81/01/29
	banner	81/01/29
	cal	81/01/29
	calendar	81/01/29
	crypt	81/01/29
	date	81/01/29
	login	81/01/29
	mail	81/01/29
	mesg	81/01/29
	passwd	81/01/29
	sara	81/01/29
	setff	81/01/29
	sort	81/01/29
	split	81/01/29
	tail	81/01/29
	who	81/01/29
	write	81/01/29
APPENDIX		
A	ASCII-tekens	81/01/29
INDEX		81/01/29

HOOFDSTUK 0 INLEIDING

Deze gebruikersgids voor systeem-commando's houdt het midden tussen een handboek en een handleiding. Het is in de eerste plaats bedoeld als een handboek: een werkje waarin men kan opzoeken hoe een bepaald commando ook al weer gegeven dient te worden. Daarnaast is getracht met wat aanvullend commentaar de (beginnende) gebruiker enigszins bij de hand te leiden, opdat hij niet al te snel vastloopt in een voor hem dan nog onbekende materie. En daarmee is deze gids tevens een handleiding. Nu kan iets niet twee dingen te gelijk zijn, en daarom gaat er een en ander verloren aan voordelen die ieder der twee aspecten heeft. Het is als "handboek" niet ideaal, en het is als "handleiding" niet ideaal. Weest u er echter van overtuigd dat het resultaat bedoeld is als een nuttig hulpmiddel bij uw werk op het Data-Entry -systeem. Mocht het daarin niet slagen, schroomt u dan niet uw kritiek kenbaar te maken bij de samenstellers.

De in deze gebruikersgids beschreven commando's vallen uiteen in vijf groepen, elk besproken in een apart hoofdstuk:

hoofdstuk 1 handelt over het bekijken en vergelijken van files, vanaf een (beeldscherm-) terminal;

hoofdstuk 2 behandelt de diverse manieren om files op papier af te drukken;

hoofdstuk 3 geeft een korte behandeling van de gegevensinvoer programma's zoals de algemene editor en de diverse specialistische editor's;

hoofdstuk 4 behandelt details van het zg. file-systeem, i.h.b. de wijze waarop files opgeborgen en teruggevonden kunnen worden, en hoe files beschermd kunnen worden tegen ongewenste manipulatie door derden;

hoofdstuk 5 behandelt een aantal, op zich staande, handige commando's, o.a. om files van en naar de SARA-installatie te sturen, te sorteren, met andere gebruikers te corresponderen, etc. .

De meeste hoofdstukken beginnen met een inleiding, daarna volgt een behandeling van de afzonderlijke commando's. Deze laatste hebben de volgende opzet:

Onder het kopje NAAM staat de naam, gevolgd door een uiterst summiere beschrijving van het commando. Het hoofdje SYNOPSIS wordt gevolgd door een beschrijving, hoe het commando vanaf de terminal gegeven moet worden. Eerst komt de naam van het commando, daarna de zg. argumenten, zoals filenamen. Argumenten die tussen "[" en "]" staan zijn niet verplicht, de andere wel. Wordt een argument gevolgd door "..." (drie puntjes), dan betekent dat, dat meerdere van die argumenten mogen volgen. In hoofdstuk 5 komt bv. het commando *write* voor, om naar een andere gebruiker te schrijven. De SYNOPSIS luidt:

`write gebruiker [ ttyr ] .`

De naam van dit commando is dus "write", het argument "gebruiker" is dus verplicht, "typr" is dat niet. En in de SYNOPSIS "`cat file ...`" betekent

"file ...", dat een of meer argumenten "file" gegeven mogen worden. Hierbij zij nog opgemerkt, dat een "file"-argument door de gebruiker van het commando wordt ingevuld met de naam van de betreffende file. Namen van files moeten altijd met een letter beginnen, en deze namen mogen het teken '*' niet bevatten. Verder komt vaak een argument van het type "[opties]" voor. Dit slaat op niet verplichte extra argumenten, die de werking van het commando beïnvloeden, zoals het aantal regels, dat een print commando per pagina af moet drukken. Opties worden nader besproken onder het kopje BESCHRIJVING, waar ook een gedetailleerde beschrijving van het commando (inclusief voorbeelden, waar nodig) te vinden is.

Eventuele verwijzingen naar andere commando's vermelden steeds tussen haakjes het nummer van het hoofdstuk waarin dat commando te beschreven wordt: bv. *write* (5).

Nodig is nog een uitleg over het woord "string", dat af en toe bij de beschrijvingen voorkomt. Het is een engels woord waarvoor we in de nederlandse taal geen equivalent hebben, althans niet in de betekenis die we hier op het oog hebben. Men duidt er een serie opeenvolgende karakters mee aan. Ieder woord is dus al een "string", maar deze zin is ook een "string". Anders dan een woord kan een "string" nl. ook een of meer *SPATIES* bevatten. Opzettelijk werd hierboven dan ook gesproken van "karakters" ipv. letters, want cijfers en leestekens doen nu ook mee: <dit is string, heus waar>, <en dit ook>, <0 57A?. u> en <\\/\//>, alsmede <.0005 e-4> vormen samen vijf verschillende string's (het doet er daarbij niet toe of men de "<" en de opvolgende ">" meerekent of niet).

Tenslotte zij opgemerkt dat de data, die links onderaan de pagina's (en tevens in de meest rechtse kolom van de inhouds-opgave) staan, aan veranderingen onderhevig kunnen zijn. Zulks ivm. mogelijke wijzigingen (cq. verbeteringen) in afzonderlijke bladzijden. Men lette er dus op dat men van elke beschrijving, die met de meest recente datering gebruikt.

HOOFDSTUK 1 BEKIJKEN EN VERGELIJKEN VAN FILES

Inleiding

In dit hoofdstuk worden enige commando's besproken, die dienstig zijn om vanaf de terminal files te bekijken, of op inhoud te onderzoeken. De beschikbare commando's zijn :

cat
cmp
diff
grep
more
spell
wc

De commando's *cat* en *more* dienen om de file-inhoud op het beeldscherm te krijgen; *cat* drukt de hele file af (alleen zinvol voor korte files), *more* maakt het mogelijk, de file "door te bladeren". Verder kan *cat* óók nog gebruikt worden om files te beschrijven.

Het commando *grep* is dienstig om snel te onderzoeken of een file (of groep van files) een bepaald stukje tekst bevat, en zo ja in welke regel(s).

De commando's *cmp* (compare) en *diff* (difference) dienen om files te vergelijken. Het commando *cmp* is bruikbaar om snel te kijken of twee files verschillen. Het commando *diff* (wat veel langzamer werkt) geeft een lijst van verschillen tussen twee files.

Tenslotte zijn er de commando's *wc* (word count) en *spell*. Hiervan kan *wc* gebruikt worden om het aantal woorden of regels in een file te weten te komen. Het commando *spell* dient als spellings-controle op engelse teksten; het meldt alle woorden die mogelijkverwijs verkeerd gespeld zijn.

NAAM

cat - (concatenate and print) print file(s) op de terminal

SYNOPSIS

cat [-] file ...

BESCHRIJVING

Cat *filenaam* stuurt de inhoud van de file "filenaam" naar de terminal, cat *filenaam1 filenaam2* ... doet dit eerst voor "filenaam1", daarna voor "filenaam2", enz.

Cat *filenaam1 filenaam2 > filenaam3* is een manier om files achter elkaar te plakken: er wordt een file "filenaam3" gecreëerd, met als inhoud die van "filenaam1", gevolgd door die van "filenaam2".

Tenslotte kan *cat* gebruikt worden om een file rechtstreeks te beschrijven, zonder tussenkomst van bv. de *editor*. Dit gaat als volgt:

cat - > *filenaam*

Nadat dit commando gegeven is, wordt alles wat ingetypt wordt op de file "filenaam" bijgeschreven, en niet verder door het systeem bekeken. Dit wordt beeindigt door *CNTL d* in te typen, waarna weer nieuwe commando's gegeven kunnen worden.

ZIE OOK

pr(4) en pt(4)

NAAM

cmp - ga na of de files verschillen (compare them)

SYNOPSIS

cmp file1 file2

BESCHRIJVING

Cmp wordt gebruikt om snel na te gaan of "file1" en "file2" verschillen. Als er geen verschil is, zegt *cmp* niets, is er wel verschil, dan verschijnt een van de volgende boodschappen op het scherm.

```
file1 file2 differ: char nnn, line mmm
EOF on file1
EOF on file2
```

De eerste boodschap betekent, dat de files verschillen, en dat het eerste verschil optreedt in het nnn-de teken van de files, dat zich in de mmm-de regel bevindt. De volgende twee boodschappen betekenen, dat "file2" begint met de inhoud van "file1", resp. "file1" begint met de inhoud van "file2". (EOF betekent end-of-file, di. het programma meldt, dat het eind van "file1" resp. "file2" gevonden is) voordat een verschillend teken gevonden is.)

ZIE OOK

diff(1)

NAAM

diff - geef de verschillen tussen twee files aan

SYNOPSIS

diff [-b] file1 file2

BESCHRIJVING

Diff verteld, wat de verschillen tussen "file1" en "file2" zijn. Dit gaat in de vorm van een lijst van wijzigingen, die op "file1" aangebracht moeten worden, om "file2" te verkrijgen. De aanwijzingen lijken op editor commando's, ze kunnen zijn:

1. *n1 a n3, n4*
2. *n1, n2 d n3*
3. *n1, n2 c n3, n4*

De betekenis is:

1. Voeg in "file1", achter regel *n1*, de regels *n3* t/m *n4* uit "file2" toe.
2. Verwijder regel *n1* t/m *n2* uit "file1". Na deze operatie, correspondeert de regel uit "file1" die nummer *n1-1* heeft, met de regel *n3*, uit "file2"
3. vervang de regels *n1* tm *n2* in "file1" door de regels *n3* tm *n4* uit "file2".

In het geval dat een traject van regels (bv. *n3, n4* in (1)) slechts uit een regel bestaat (als dus *n3 = n4* in (1)), wordt maar één regelnummer gegeven (bv. *n1 a n3*).

Na elk vermeld verschil volgen steeds de betreffende regels uit beide files; eerst die uit "file1", gemarkeerd door een "*", daarna die uit "file2", gemarkeerd door een ".".

De -b optie dient er voor om *SPATIES* en *TABs* aan het eind van de regel (die zijn niet zichtbaar) uit te sluiten van het vergelijken. Ook de string's die uitsluitend uit *SPATIES* bestaan worden niet meegenomen.

ZIE OOK

cmp(1)

NAAM

grep - kijk of een file bepaalde regel(s) bevat

SYNOPSIS

grep [opties] string file

BESCHRIJVING

Met *grep* kunt u snel onderzoeken, of een file regels bevat, die op hun beurt een bepaalde string bevatten. Alle dergelijke regels worden op de terminal afgedrukt.

Bv. *grep 'MC' jaarverslag* stuurt alle regels uit de file "jaarverslag", die de string "MC" bevatten, naar de terminal. Het "string"-argument bestaat dus hier uit de eigenlijke tekst, tussen aanhalingstekens, maar er is veel meer mogelijk:

De "." in een "string"-argument staat voor een willekeurig teken, *grep 'M.' jaarverslag* geeft alle regels die "MC", "Mc", "Ml", etc. bevatten.

De "*" staat voor een willekeurige herhaling van de voorgaande teken: *'ho*p'* geeft de teksten "hop", "hoop", "hooop", etc. aan. Tekens tussen "[" en "]" in een "string"-argument betekenen "naar keuze" een van deze tekens. *'b[aeio]k'* slaat op de teksten "bak", "bek", "bik", "bok", maar niet bv. "buk".

Ook is het gebruik van uitdrukkingen als *[b-m]* toegestaan, met als betekenis: de tekens "b" t/m "m" (zie Appendix A voor volgorde van karakters).

Nog een voorbeeld:

grep 'St.[stuv]m' jaarverslag* zoekt alle regels in "jaarverslag", die een string bevatten, beginnend met "St", dan een willekeurig stuk tekst (willekeurig veel willekeurige tekens via ".*"), dan een "s" of "t" of "u" of "v" en tenslotte een "m". Een voorbeeld van zo'n tekst is "Stichting Mathematisch Centrum".

De opties zijn:

- v druk alleen die regels af die de opgegeven string niet bevatten;
- c alleen het aantal regels, die de opgegeven string bevatten, wordt afgedrukt;
- n elke afgedrukte regel wordt voorafgegaan door zijn regelnummer.

NAAM

`more` - blader een file door

SYNOPSIS

`more` [opties] file ...

BESCHRIJVING

More is een programma om een file door te lezen respectievelijk te doorzoeken. Dit is vooral nuttig bij redelijk grote tot grote files. Na het afdrukken van een pagina tekst stopt *more* met printen en geeft de boodschap "--More--" onderaan het beeldscherm. Door het intypen van een *SPATIE* krijgt de gebruiker de volgende pagina te zien. Als deze *SPATIE* vooraf wordt gegaan door een getal (bv. 5) dan krijgt men de eerstvolgende vijf regels te zien. Het intikken van *CNTL d* heeft tot gevolg dat er 11 extra regels worden afgedrukt, d.i. een half scherm vol.

More kijkt op wat voor type terminal u aan het werk bent en bepaald aan de hand daarvan de lengte van zijn pagina's.

Naast *SPATIE* en een getal gevolgd door een *SPATIE* kunnen ook andere aanwijzingen ingetypt worden (daarbij is "N" een door de gebruiker willekeurig te kiezen getal):

Nz hetzelfde als het intypen van een *SPATIE*, maar nu wordt N de nieuwe pagina lengte;

Ns skip N regels en druk dan een pagina af;

Nj skip N pagina's en druk dan een pagina af;

Nn skip naar N-de file uit de commando-regel (indien N een niet zinvolle waarde heeft, dan wordt de laatste file uit de serie genomen);

Np skip naar de N-de voorgaande file uit de commando regel (indien u bezig bent in een bepaalde file dan gaat *more* eerst terug naar het begin van die file waaraan u bezig bent;

q verlaat *more* ;

N/S zoek tot het N-de voorkomen van de string "S" van karakters gevonden is; als er geen N voorkomens van "S" meer zijn dan blijft de situatie zoals die was.

De commando's treden onmiddellijk in werking, dit houdt in dat het niet nodig is een *RETURN* in te typen. Totdat het laatste karakter is gegeven kunt u gebruik maken van de *CANCEL LINE* toets.

De output kan te allen tijde onderbroken worden door het intypen van *CNTL * ; *more* zal dan stoppen met afdrukken en terugkomen met de "--More--" boodschap.

De opties:

- d in plaats van "--More--" print het programma de boodschap: "Hit space to continue, Rubout to abort;"
- N brengt de paginalengte op N regels ipv. de standaard maat.
- +N start op de N-de regel ipv. de eerste.
- +/S begin twee regels vóór het eerstvolgende voorkomen van de string "S".

NAAM

`spell` - signaleer (mogelijke) spelfouten in engelse tekst

SYNOPSIS

`spell` [opties] file ...

BESCHRIJVING

Spell zoekt de woorden uit de genoemde files op in een *engelstalige* woordenlijst. Woorden die daar niet in voorkomen, en niet afleidbaar zijn uit voorkomende woorden (bv. door er achtervoegsels als "ing" bij te plaatsen), worden vermeld.

Opties zijn *-v* en *-b*.

-v woorden die niet letterlijk in de woordenlijst voorkomen worden alle vermeld; is het woord wel afleidbaar uit een voorkomend woord, dan wordt dat apart vermeld;

-b Er wordt gecontroleerd op strikt *engelse* spelling, zonder deze optie is dit *engels-amerikaans*. Het is niet verplicht, filenamen als argument mee te geven, doen we dit niet, dan controleert *spell* de vanaf de terminal ingetypte tekst, totdat *CNTL d* getypt wordt. Dit is handig om even te controleren, of een bepaald woord gespeld wordt, zoals de gebruiker denkt, bv.

```
spell
tabel
table
CNTL d
tabel
```

Het *spell* commando wordt hier gegeven zonder invoer file, en de woorden *table* en *tabel* worden ter controle aangeboden. *CNTL d* geeft hier aan, dat er geen verdere invoer voor *spell* komt. *Spell* antwoord met "tabel", een woord dat hij niet kent.

NAAM

wc - tel woorden (regels) in een file (word count)

SYNOPSIS

wc [opties] file ...

BESCHRIJVING

Wc print het aantal woorden , zowel als het aantal regels dat voorkomt in de files wiens namen als argument gegeven worden. Als "woord" wordt elke serie tekens genomen, begrensd door *SPATIES*, *TABS* of *CARRIAGE RETURN* symbolen.

Er zijn twee opties:

- l print alleen het aantal regels.
- w print alleen het aantal woorden.

"this page is left blank intentionally"

HOOFDSTUK 2 PRINT-PROGRAMMA'S

Inleiding

Het betreft hier de commando's :

hqp
lpr
pr
print
pt
qupr
rdcas
wrcas

De eenvoudigste manier om een file te printen is, dmv. het *print* commando. Zo resulteert *print filenaam* in het afdrukken van de file met de aangegeven naam. Beschikt u zelf over een printer, dan wordt die gebruikt, anders de centrale printer. In het laatste geval, krijgt u via "mail" bericht als het afdrukken klaar is. Het commando *print* drukt af, in de vorm van een-koloms, genummerde pagina's, voorzien van een kop. Iha. zal het printen niet onmiddellijk plaats vinden. Er wordt door het systeem nl. per printer een lijst bijgehouden van files die afgedrukt moeten worden. Wordt een print commando gegeven, terwijl deze lijst niet leeg is, dan moet op de beurt gewacht worden. Ondertussen kan wel vanaf de terminal aan andere dingen doorgewerkt worden.

Bent u niet tevreden met de standaard print methode, dan zijn er de commando's *pr* en *pt*. Hiermee heeft u verregaande controle over het print formaat, di. het aantal kolommen per pagina, het aantal regels per pagina, het aantal letters per kolom, wel of niet een kopje, etc. Noch *pr* en noch *pt* zorgen er echter voor dat de door hen bewerkte file in de wachtlijst voor een printer terecht komt: de afdruk komt op het scherm. Moet er daadwerkelijk op papier geprint worden, dan moet *pr* of *pt* gebruikt worden in combinatie met *lpr*, die de uitvoer van *pr* (*pt*) in een printer-wachtlijst plaatst, bv.

pt -a -2 -150 filenaam | lpr
roept eerst *pt* aan, met als uitvoer de text van "filenaam", over twee kolommen (-2), met een kopje boven elke pagina (-a) en 50 regels per pagina (-150). Deze uitvoer verschijnt niet op het scherm, maar wordt middels *lpr* doorgestuurd naar een printer.

Beschikt u over een HCT300 terminal of iets dergelijks, dan is er de mogelijkheid om afdrukken te maken die zeer geschikt zijn voor offset. Hiertoe dient u te printen met het commando *hqp* (high quality print), dat extra zwart afdrukt, u de gelegenheid geeft na elke pagina het papier opnieuw precies in te stellen, etc. .

Tenslotte: indien het wenselijk is files in lettertekens uit twee verschillende alfabetten af te drukken (bv. gewoon en cursief), dan kan dit op de QUME printer via het commando *qupr* . Deze printer bevindt zich in de printerruimte op de derde verdieping. Dit speciale commando mag alleen gegeven worden vanaf de terminal die naast deze printer staat. Neem daarbij echter eerst even contact op met een van de systeembeheerders.

NAAM

hqp - kwaliteits print commando voor HCT en HYPRINT terminals

SYNOPSIS

hqp [opties] file ...

BESCHRIJVING

Hqp is speciaal geschreven voor die terminals die een HQP schakelaar hebben, dat zijn bij ons de HCT-302 en DIABLO-HYPRINT printer/-terminals. Als deze schakelaar omgezet is wordt ieder karakter twee keer op dezelfde plaats afgedrukt; de zwarting is daardoor dan beter. Zorg er echter voor dat deze HQP schakelaar na gebruik van het *hqp* commando weer uitgezet wordt.

Hqp stelt TAB-posities in op ieder veelvoud van 8 SPATIES en de linker kantlijn begint op positie 8, zodat nieuw papier gemakkelijk ingedraaid kan worden. Inkomende boodschappen (van *mail* (5) etc.) worden geweerd. Het is optioneel mogelijk halve LINEFEED's tussen de regels in te voegen, zodat de regelafstand anderhalve regel wordt. *Hqp* accepteert pagina's die mv. FORMFEED's onderscheiden zijn; dit mag gewijzigd worden in een ander karakter, bv. een "*" of een "-", of door het bereiken van een bepaald aantal regels (zie de opties).

De terminal zal aan het einde van iedere pagina stoppen en een pieptoon geven; na het geven van een RETURN gaat het programma weer door met printen. Als hier een SPATIE wordt getypt dan wordt de volgende pagina geskipt, wordt er een LINEFEED ingetypt dan resulteert dat in een REVERSE LINEFEED (dit is één LINEFEED terug) en daarna het printen van de pagina. Alle andere ingetypte karakters worden niet verwerkt. Er zijn verscheidene opties mogelijk:

- fC er wordt begonnen aan een nieuwe pagina als het karakter "C" wordt aangetroffen ("C" is een door de gebruiker willekeurig te kiezen karakter, standaard is de FORMFEED);
- h er wordt na iedere regel een HALF LINEFEED toegevoegd;
- pN een nieuwe pagina wordt gestart na N regels, de optie -p0 schakelt het paginerings systeem uit.

ZIE OOK

HCT handleiding , op aanvraag bij de systeem-beheerders;
qupr(2)

NAAM

lpr - print programma voor de LA-120 matrix printer

SYNOPSIS

lpr [optie] file ...

BESCHRIJVING

Lpr stuurt één of meerdere files naar een LA-120 matrix printer. *Lpr* zet standaard uw naam in *banner* (5) -letters op het papier met een datum en tijd. Als de file(s) geprint zijn wordt dat gerapporteerd door u een bericht te sturen met behulp van *mail* (5).

De volgende opties zijn aanwezig:

- r verwijder de file direct nadat hij aangeboden is,
- c copieer de file, zodat de file zelf weer gemodificeerd kan worden zonder dat dit invloed heeft op de uitvoer van *lpr* (er kan enig tijdsverloop zijn tussen het geven van een print-opdracht en de uitvoering ervan, terwijl u misschien verder wilt gaan met het aanbrengen van veranderingen via *editor* od.) .
- n geef geen boodschap mbv. van het *mail* commando,
- m stuur de boodschap wel (standaard),
- b druk de *banner* (5) -tekst niet af,
- pM gebruik printer M , waarbij
 - M=1: de printer in de computer ruimte;
 - M=2: de printer bij de "Dienst Informatie";
 - M=3: de printer van de ponskamer.Bij weglating van deze optie gaat de uitvoer naar de eigen groepsprinter.

ZIE OOK

hqp(2), *pr*(2), *print*(2), *pt*(2)

NAAM

pr - print file

SYNOPSIS

pr [opties] file ... [[opties] filek ...]

BESCHRIJVING

Pr drukt de inhoud af van één of meerdere files. De uitvoer wordt verdeeld in pagina's met als kop de datum, de naam van de file, en het pagina nummer. Bij ontbreken van "file"-argumenten wordt de uitvoer direct naar de terminal geschreven. Bij meerdere "file"-argumenten zijn de opties van toepassing op al die argumenten, maar zij mogen tussen twee opeenvolgende files opnieuw gezet worden.

In de onderstaande beschrijving van de opties wordt met "N" een willekeurig te kiezen getal bedoeld. De opties zijn :

- c druk in de kopregel de filenaam in dikke letters af (dit kan alleen op de LA-120 matrix printer !);
- hS gebruik de string "S" als koptekst;
- lN maak de pagina-lengte N regels ipv. de standaard van 66 regels;
- m print de opgegeven files naast elkaar, ieder in een aparte kolom;
- sC separeer de kolommen dmv. het karakter "C", in plaats van ze uit te vullen met *SPATIES*. Als C (een door de gebruiker willekeurig te kiezen karakter) niet wordt meegegeven wordt hiervoor een *TAB* gebruikt;
- t print de 5 regels kostende kopregel niet, evenals de 5 lege regels die aan het eind van de pagina komen;
- p-v onderdruk het printen van de tekst " MC Data-Entry " in de kopregel;
- wN neem bij uitvoer over meerdere kolommen een pagina-breedte van N karakter-posities (ipv. het standaard aantal van 72) in acht;
- N druk af in N kolommen;
- +N begin bij de N-de pagina.

Het schrijven naar uw terminal door andere gebruikers, dmv. het *write* (5) commando, wordt gedurende het printen voorkomen.

ZIE OOK

hqp(2), lpr(2), print(2) en pt(2)

NAAM

print - print een file

SYNOPSIS

print file ...

BESCHRIJVING

Print is het standaard print commando voor de LA-120 matrix printers. Het programma drukt één of meer files door middel van het *pt* en het *lpr* commando. De file(s) wordt afgedrukt op de printer die bij uw groep hoort.

ZIE OOK

hqp(2), lpr(2), pr(2) en pt(2)

NAAM

pt - print een file

SYNOPSIS

pt [opties] file ... [[opties] filek ...]

BESCHRIJVING

Pt drukt de inhoud af van één of meerdere files. De uitvoer wordt verdeeld in pagina's met als kop de datum, de naam van de file, en het pagina nummer. Bij ontbreken van "file"-argumenten wordt de uitvoer direct naar de terminal geschreven. Bij meerdere "file"-argumenten zijn de opties van toepassing op al die argumenten, maar zij mogen tussen twee opeenvolgende files opnieuw gezet worden.

In de onderstaande beschrijving van de opties wordt met "N" een willekeurig te kiezen getal bedoeld. De opties zijn :

- a print de filenaam in de kopregel in dikke letters (dit kan alleen op de LA-120 matrix printer !);
- c print het "Data-Entry" in de kopregel;
- f geef geen *FORMFEED* na iedere pagina, doch vul de pagina uit met lege regels;
- hs gebruik de string "S" als koptekst;
- k geef de filenaam, onder de kopregel, in *banner* (5) vorm;
- lN neem de lengte van iedere pagina *N* regels lang, in plaats van de standaard 66 regels (bij weglating van "N" wordt de waarde "1" gehanteerd);
- n geef regelnummers erbij;
- p druk de naam van de gebruiker er niet bij;
- t print de 5 regels kostende kopregel niet, evenals de 5 lege regels die aan het eind van de pagina komen.
- v onderdruk het printen van de tekst " 11/34 " in de kopregel;
- wN verander de standaard regellengte (72) in "N";
- +N begin met de N-de pagina;

Het schrijven naar uw terminal door andere gebruikers, dmv. het *write* (5) commando, wordt gedurende het printen voorkomen.

ZIE OOK

hqp(2), lpr(2), pr(2) en print(2)

NAAM

qupr - print file op de QUME printer

SYNOPSIS

qupr [opties] file ...

BESCHRIJVING

Qupr drukt tekstfiles op de QUME S3-TwinTrack printer af. Dat is een tweekops HCT, waarvan de twee printkoppen dmv. letterwielletjes voorzien kunnen worden van twee verschillende soorten alfabetten (bv. "gewoon" en cursief). De output van *qupr* komt altijd op de QUME terecht. De files mogen de volgende karakters bevatten:

1. alle printbare ASCII karakters (zie Appendix A)
2. de normale besturings-karakters, zoals *NEWLINE* , *BACKSPACE* , *CARRIAGE RETURN* , *FORMFEED*
3. de speciale besturings-karakters *SO* (*CNTL n*) om het rechter letterwielletje en *SI* (*CNTL o*) om het linker letterwielletje te specificeren (standaard).
4. speciale paren tekens die beginnen met *ESC* (escape):
ESC 7 : *REVERSE LINEFEED* (één *LINEFEED* terug)
ESC 8 : *REVERSE HALF LINEFEED*
ESC 9 : *HALF LINEFEED*
ESC - : *REVERSE QUARTER LINEFEED*
ESC + : *QUARTER LINEFEED*

Alle andere karakters en escape sequences (dit zijn opeenvolgingen van een *ESC* plus één karakter) worden geskipt. Lege pagina's worden geskipt.

De opties die meegegeven kunnen worden ("N" is hierbij een door de gebruiker willekeurig te kiezen getal) zijn :

- sN stop na het afdrukken van N regels. Standaard komt N overeen met de paginalengte.
- pN paginalengte in regels.
- lN aantal regels per inch, standaard: N=6
- cN aantal karakters per inch, standaard: N=12
- oN zet de linkerkantlijn N *SPATIES* vanaf de meest linkse positie. Standaard is de linkerkantlijn 5.5" rechts van de meest linkse positie, waardoor het gehele gebied door beide letterwielletjes bestreken kan worden. Het meegeven van de optie -o0 geeft u het gebruik van de volledige papierbreedte, maar dat kan alleen bij

gebruik van het linker letterwielkje;

- 2 gebruik de beide letterwielktes om twee verschillende alfabetten door elkaar te gebruiken: standaard wordt alleen het linker letterwielkje gebruikt; bij het tegenkomen van een *SI* karakter wordt overgegaan op het rechter, waarna weer teruggegaan wordt bij het *SO* karakter;
- b print niet heen en weer: standaard zal de printkop tijdens het printen afwisselend van links naar rechts en van rechts naar links typen, zulks om de printsnelheid te optimaliseren (voor hoge kwaliteit is dit echter ongeschikt en dient dus onderdrukt te worden via deze optie).

ZIE OOK

Handleiding formatter

NAAM

`rdcas` - copieer een cassette in een file

SYNOPSIS

`rdcas` file

BESCHRIJVING

Het `rdcas` commando copieert een cassette naar een file. Indien de file-naam al in de betreffende directory voorkomt volgt eerst een vraag of de bestaande file overschreven mag worden.

ZIE OOK

`wrcas(2)`

NAAM

wrcas - schrijf een of meer files op een cassette

SYNOPSIS

wrcas file ...

BESCHRIJVING

Het *wrcas* commando copieert een of meer files naar een cassette ; een cassette zonder schrijfpermissie (het bekende palletje is eruit) kan niet beschreven worden.

Als meerdere files op één cassette worden geschreven, dan worden die files eerst samengevoegd tot een geheel en daarna op de cassette gezet.

ZIE OOK

rdcas(2)

"this page is left blank intentionally"

HOOFDSTUK 3 GEGEVENSINVOER-PROGRAMMA'S

Hoewel het computer-systeem van de PDP11/34 systeem-commando's kent waarmee gegevens-invoer bedreven kan worden, zullen die noch in dit hoofdstuk en noch in deze gids aan de orde komen. Wel aan de orde zullen komen die programma's die speciaal voor Data-Entry op het MC beschikbaar zijn gemaakt (of dat nog zullen worden). De hier te behandelen stof valt dus in eigenlijke zin niet onder de in deze gids te bespreken systeem-commando's (dwz. bij het computer-systeem behorende commando's). Dat deze stof toch hier te vinden is komt voort uit de overweging dat het voor Data-Entry-gebruiker handig zal zijn in één publicatie zoveel mogelijk details te kunnen opzoeken.

De van toepassing zijnde programma's,

bibed (nog in voorbereiding)
concept
ded
editor

zullen in dit hoofdstuk zo kort mogelijk behandeld worden. En wel in de vorm van overzichten van de aanwijzingen die de gebruiker aan de programma's kan geven. In de eerste plaats is er het standaard invoer-programma *editor*, geschikt voor willekeurige teksten. Voor het gecontroleerd invoeren van gegevens die aan een ponsconcept moeten voldoen, is er het programma *ded* en het bijbehorende programma *concept* om concepten aan te maken. Tenslotte is er het speciaal voor bibliotheek-toepassingen (titel beschrijvingen) geschreven programma *bibed* (nog in voorbereiding). Al deze programma's worden hier slechts in telegramstijl besproken. Voor uitvoeriger beschrijvingen raadplege men:

1. *Handleiding bibliotheek-editor* (nog in voorbereiding);
2. *Handleiding ponsprogrammatuur* ;
3. *Handleiding editor* .

NAAM

concept - maak een ponsconcept aan

SYNOPSIS

concept [-]

BESCHRIJVING

Voor een gedetailleerde beschrijving van *concept* raadplege men de "*Handleiding ponsprogrammatuur*". De aanroep "*concept* - " wordt gebruikt om een reeds bestaand concept te inspecteren; met *concept* kunnen nieuwe concepten worden aangemaakt. Het programma werkt interactief - op elk moment worden naar de noodzakelijke gegevens gevraagd - om een concept op te bouwen. De gebruikers-antwoorden kunnen bestaan uit de volgende elementen:

*	alles
@	alfabetisch
#	alfa + cijfers
1	cijfers
a	letters
Z	hoofdletters
z	kleine letters
+	een + of - teken
n	cijfer of . , + - e
.	uitgevulde rubriek
d	dupliceer
-	automatische skip
!	scheider tussen aangrenzende rubrieken van dezelfde soort
#	skip niet toegestaan voor corresponderende rubriek

Toegestane uitgevulde rubriek specificaties:

centrering

>a	plaats rechts, vul uit met a
<a	plaats links, vult uit met a
ab	plaats in het midden, vul links uit met a en rechts met b

Types veldbeschrijving:

x
[x]
[x,n,m]
[x,n,.]
xy
x[y]
x[y,n,m]
x[y,n,.]

Al deze uitdrukkingen mogen voorafgegaan en gevolgd worden door:

("symbolen")

Een x of y stellen hierbij, hetzij één der eerste zeven symbolen uit paragraaf 5.1 (van "*Handleiding ponsprogrammatuur*"), hetzij zo'n symbool voorafgegaan door een "\", hetzij een enkel ander symbool voor.

NAAM

ded - data-entry editor

SYNOPSISdedBESCHRIJVING

In het volgende overzicht stelt "nn" een getal voor.

nn	vertoon regel met nummer "nn"
+nn	vertoon de regel die "nn" regels verder ligt
-nn	vertoon de regel die "nn" regels terug ligt
.	vertoon de huidige regel nogmaals
+.nn	hetzelfde als +nn
-.nn	hetzelfde als -nn
\$	vertoon de laatste regel
\$-nn	vertoon de regel die "nn" vóór de laatste ligt
x,y	vertoon regels x tm y, waarbij x en y op een van de boven vermeldde wijzen regelnummers aangeven
l/S/	(of l//), vertoon de eerstvolgende regel die de string in "S" bevat
=	druk het nummer af van de huidige regel
a	voeg tekst toe aan het eind van het bestand
i	voeg tekst in vóór de huidige regel
c	controle ponsen, vanaf de huidige regel
.	einde van achtervoegen, invoegen of controle ponsen
v	verbeter de huidige regel
w	gooi de huidige regel weg
nn,mmw	gooi regel nn t/m mm weg
d	elke regel dient meteen twee keer geponst te worden
e	elke regel wordt slechts een keer ingevoerd, afgezien van eventueel later controle ponsen
k	maak een kopie van de huidige regel voor latere duplicatie
s	stop het "ded" programma
!nn	concept-keuze, gebruik tot nader order concept nn
>	skip over rubriek of beeindig uitgevulde rubriek
~	beeindig deze regel
CNTL x	gooi de invoer van de huidige regel weg
BACKSPACE	spring een rubriek terug in de huidige regel
CNTL v	dupliceer vanaf de vorige regel
CNTL k	dupliceer vanaf de eerder aangemaakte kopie
CNTL c	dupliceer vanaf de laatste controle-regel
^	zet de controle af voor de rest van de huidige rubriek
CNTL a	verandering van dupliceer positie gewenst, voorafgaand aan eerstvolgende duplicatie
+	beweeg dupliceerpositie een plaats naar rechts
-	beweeg dupliceerpositie een plaats naar links
RETURN	duplicatie positie goed ingesteld, dupliceer nu

Opmerking: de commando's a, i, c, v, w, d, e, k, s en diegene waarbij de CNTL -stoets gebruikt wordt, mogen ook met hoofdletters getypt worden.

NAAM

editor - "edit" tekstinvoer

SYNOPSIS**editor**BESCHRIJVING

Overzicht editor-commando's:

default parame- ter(s)	max nr par's	opdracht id- entificatie met postpar	bete- kenis	current line na uitvoering
.	1	a	append	laatst toegevoegde regel
-	0	b	brief	geen prompt symbolen c: en l:
.,pl	2	d{v}	delete {veto}	regel volgend op de laatst verwijderde regel
-	0	e <file>	edit	eerste regel
-	0	f	filenaam	
.	1	g	glue	de nieuw gevormde regel
l	1	l/<string>/	locate forward	de eerste regel na of gelijk aan de specificeerde met de <string>
\$	1	l?<string>?	locate backward	de eerste regel voor of gelijk aan de gespecificeerde met de <string>
.,pl	2	m <n>	move	de laatst verplaatste regel
.,pl	2	p	print	de laatst afgedrukte regel
-	0	q	quit	
.	1	r <file>	read	
.,pl	2	s{g}/<str1>/<str2>	substitute {global}	de laatste regel, waarin een een verandering is aangebracht
-	0	v	verbose	herstel de prompt symbolen

1,\$	2	w <file>	write	
.,pl				
\$	1	=	regelnr	
.+1	1	<leeg>	getline	de gespecificeerde regel

=====

Karakters met een speciale betekenis bij

s: ^ begin van de regel, \$ eind van de regel, * range,
 & herhalings-symbool, \ escape symbool ;

par^s: \$ laatste regel, . current line;

Toelichting. Een .,pl in de kolom 'default parameter(s)' duidt aan:
als er geen parameter wordt gespecificeerd, dan is . (current line) de
eerste parameter;

als er minder dan twee parameters gespecificeerd zijn, dient de tweede
parameter als gelijk aan de eerste beschouwd te worden.

HOOFDSTUK 4 HET FILE-SYSTEEM

Inleiding

In dit hoofdstuk worden de volgende commando's besproken :

cd
chdir
chmod
cp
ls
mkdir
mv
pwd
rm
rmdir

Files (gegevensbestanden) worden in het systeem opgeborgen en teruggevonden onder een naam, de zg. filenaam . Dit gaat per gebruiker (eigenlijk per login-naam). Het systeem houdt nl. voor elke gebruiker (login-naam) een zg. directory (naslag lijst) van de files van die gebruiker bij. Wil een gebruiker zo'n file bv. "editen", dan wordt alleen in de directory van die gebruiker naar die file gezocht. Het gevolg is, dat verschillende gebruikers dezelfde naam voor een file kunnen kiezen, zonder dat verwarring ontstaat; de namen van deze files komen nl. in verschillende directories.

Elementaire operaties op files kunnen worden uitgevoerd door de commandos : *cp* , *ls* , *mv* en *rm* . Het commando *cp* wordt gebruikt om files te copieëren; er ontstaan dan meerdere files met verschillende naam maar dezelfde inhoud. Het commando *ls* (list) kan gebruikt worden om te bekijken, welke files de gebruiker heeft (di. welke in de betreffende directory zitten). In zijn eenvoudigste vorm drukt *ls* de namen af van alle files in een directory. Het is ook mogelijk extra informatie te krijgen, zoals de grootte van een file, het tijdstip waarop hij het laatst gewijzigd is, etc. Het *mv* (move) commando kan gebruikt worden om een file onder een andere naam op te bergen. Met *rm* (remove) kan men files uit het systeem verwijderen.

Een ander aspect van het file-systeem is de mogelijkheid files te beschermen tegen gebruik (lezen, schrijven) door anderen. Elke gebruiker is ingedeeld in een bepaalde groep. (Er is bv. een groep "ponskamer", "bibliotheek", etc.). Een gebruiker kan nu voor ieder van zijn files apart vaststellen of hij gelezen, resp. beschreven mag worden alleen door hem/haar zelf, de leden van zijn groep, of alle gebruikers. Dit gaat met het commando *chmod* (change mode).

Tot dusverre is er vanuit gegaan, dat elke gebruiker slechts één standaard-directory heeft. Dit hoeft niet altijd zo te zijn, aangezien elke directory zg. subdirectories kan bevatten, die via deze directory toegankelijk zijn. Stel dat we een groep files, die tot één project behoren, ordelijk bij elkaar willen houden. Dit kan als volgt. Nadat we zijn "in-

gelogd", bevinden we ons automatisch in de voor ons geldende standaard-directory. Stel nu dat we de serie files willen aanmaken, samengegroept onder de naam "projektl". Het commando *mkdir projektl* maakt nu een directory (naslaglijst) met de naam "projektl" aan. Geven we vervolgens het commando *chdir projektl* (change directory), dan verlaten we onze standaard-directory en komen in de directory "projektl" terecht. Hierin kunnen we nu, bv. met *editor*, onze files aanmaken. Voeren we het *ls* commando uit, dan krijgen we nu ook alleen die files te zien, die in directory "projektl" aangemaakt zijn. Voeren we daarentegen *ls* uit, terwijl we in de standaard-directory zitten, dan wordt wel gemeld, dat de standaard-directory de subdirectory "projektl" bevat (een directory is nl. zelf ook een file), maar de files die onder "projektl" zitten, worden niet vermeld. Vanuit de standaard-directory zijn de files in "projektl" dan ook niet zonder meer toegankelijk; ofwel er moet eerst *chdir projektl* gegeven worden, ofwel er moet aan een file, zeg van naam "xxx" gerefereerd worden als "projektl/xxx".

[Opmerking: waar u met de naamgeving aan files (en directories) wel aan moet denken is, dat namen van files (en directories) moeten beginnen met een letter, en dat deze namen mogen niet het karakter '*' bevatten].

Dit principe van subdirectories is erg flexibel en kan dan ook gemakkelijk uitgebreid worden, al lijkt dat in het begin erg ingewikkeld. Binnen een subdirectory kunnen nl. weer nieuwe subdirectories aangemaakt worden, die ook weer via *chdir* bereikbaar zijn. Willen we vanuit een (sub)directory terug naar een directory die een niveau hoger ligt, dan gaat dit met *chdir ..*. Vanuit "projektl" uit het voorbeeld zou dit commando ons terugbrengen naar de standaard-directory van de gebruiker. Om te zien in welke directory we nu zitten (mochten we dat vergeten zijn) kan het commando *pwd* (print working directory) gebruikt worden. Indien bv. een gebruiker "yy", die tot de groep "bibliotheek" behoort, meteen na "inloggen" *pwd* intypt, verschijnt op het scherm de tekst *"/bibl/yy"*. "bibl" is een directory waarin zich de standaarddirectories (hier "yy") van alle leden der groep bibliotheek bevinden. Na een *chdir projektl*, zou *pwd* antwoorden *"/bibl/yy/-projektl"*. Wordt hierna echter *chdir ..* gedaan, dan zitten we weer in *"/bibl/yy"*.

Een van de voordelen van het gebruik van subdirectories is, dat in verschillende subdirectories, verschillende files van dezelfde naam voor kunnen komen. Hierdoor kunnen filenamen kort gehouden worden, terwijl toch veel verschillende files aanwezig zijn. Voorbeeld: de reproductie-dienst zou per rapport dat in bewerking is, een subdirectory met als naam (een afkorting van) de titel van dat rapport kunnen aanmaken. Binnen zo'n directory kunnen dan de filenamen "hfdst1", "hfdst2", etc. gebruikt worden, zonder dat verwarring met de hoofdstukken van andere rapporten mogelijk is. Tenslotte: net zo als gewone files verwijderd worden door *rm*, kan men directories verwijderen met *rmdir*.

CD ga naar een ander directory

1

NAAM

cd - ga naar een ander directory (change directory)

SYNOPSIS

cd directory

BESCHRIJVING

Zie bij *chdir* (4), want *cd* is een verkorte schrijfwijze voor *chdir*.

NAAM

chdir - ga naar een ander directory (change directory)

SYNOPSIS

chdir directory

BESCHRIJVING

Als de huidige directory een subdirectory bevat, dan kan deze subdirectory met *chdir* tot de huidige gemaakt worden. Is "sdir" zo een subdirectory dan kan dat mbv. de commando-regel *chdir sdir* . Door middel van *chdir ..* komen we dan vanuit de subdirectory weer terug in de overkoepelende directory.

Stel, we hebben een directory "dir" die een subdirectory "sdir1" bevat, die op zijn beurt weer een subdirectory "sdir2" bevat, welke weer een subdirectory "sdir3" bevat. We kunnen dan van "dir" in "sdir3" terecht komen, en wel door na elkaar de commando's *chdir sdir1* , *chdir sdir2* en *chdir sdir3* te geven. Dit kan echter gelukkig ook in één keer, nl. met *chdir sdir1/sdir2/sdir3* . We kunnen terug, door driemaal *chdir ..* te geven, of via *chdir ../..* .

Het commando *cd* kan overal ipv. *chdir* gebruikt worden, het is slechts een verkorte schrijfwijze.

NAAM

`chmod` - verander lees- of schrijfpermissie van een file (change mode)

SYNOPSIS

`chmod` permissies file ...

`chmod` permissies directory ...

BESCHRIJVING

Voor elke file (cq. directory) kan de eigenaar vaststellen, of deze gelezen, resp. beschreven mag worden door de eigenaar, door de groep waartoe deze eigenaar behoort en/of door alle andere gebruikers (derden). Standaard worden de files aangemaakt met lees- en schrijfpermissie voor de eigenaar en diens groep, en alleen leespermissie voor derden. Indien dit niet volgens wens is, moet *chmod* gebruikt worden.

Het argument "file" duidt de file aan, waarvan permissies veranderd moeten worden, en in het "permissies"-argument wordt aangegeven hoe.

De letters "u" (user), "g" (group) en "o" (others) worden gebruikt om aan te geven of de permissies van de eigenaar, van de groep of van derden veranderd dient te worden. Bovendien kan met "a" (all) aan alle gebruikers gerefereerd worden. Permissies worden genoteerd met "w" (write) voor schrijfpermissie en "r" (read) voor leespermissie. Permissies worden gegeven met "+", ingetrokken met "-" en geheel opnieuw vastgesteld met "=".

Voorbeelden:

chmod o+w file Derden krijgen schrijfpermissie op de betreffende file, permissies blijven verder ongewijzigd.

chmod g-r file Trek de leespermissies voor groepsleden in, andere permissies veranderen niet.

chmod g=r file De groepsleden krijgen leespermissie.

chmod ug=wr file De permissie voor de eigenaar en de groepsleden wordt: lezen en schrijven.

chmod a+r,o-w file Een ieder krijgt leespermissie, eventuele schrijfpermissie voor derden wordt ingetrokken.

ZIE OOK

`crypt(5)`, `ls(4)`, `mkdir(4)` en `rmdir(4)`

NAAM

cp - copiëer

SYNOPSIS

cp file1 file2
cp file ... directory

BESCHRIJVING

In de eerste aanroepsvorm wordt "file1" gecopiëerd naar "file2" . Er ontstaat dus een tweede file, met dezelfde inhoud, maar afwijkende naam. Als "file2" al eerder bestond, wordt hij overschreven, zijn oorspronkelijke inhoud raakt dan verloren. Indien "file2" nog niet bestond, wordt zijn eigenaar dezelfde als die van "file1", tevens worden de permissies voor "file1" (lezen/schrijven voor eigenaar, groep en anderen) overgenomen. Bestond "file2" al, dan blijven zijn eigenaar en permissies ongewijzigd. In de tweede aanroepsvorm worden een of meer les (file ...) naar de directory gecopiëerd. Bv: *cp file1 file2 dir* leidt ertoe, dat de copieën van "file1" en "file2" in de directory "dir" terecht komen, ook weer onder naam "file1" en "file2" (er ontstaan dus "dir/file1" en "dir/file2", naast "file1" en "file2" in de huidige directory, die blijven bestaan).

ZIE OOK

cat(1), pr(2) en mv(4)

NAAM

`ls` - geef de inhoud van een directory weer (list)

SYNOPSIS

```
ls [ opties ] [ directory ]
ls [ opties ] [ file ... ]
```

BESCHRIJVING

`Ls` ,zonder argumenten, geeft de namen van alle files en subdirectories die bevat zijn in de huidige directory . Wordt als argument een directory-naam meegegeven, dan wordt die gebruikt ipv. de huidige directory. Er zijn vele opties (waarvan `-l` verreweg de belangrijkste is) :

- `-l` Geef niet alleen de filenamen , maar voor elke genoemde file de naam van zijn eigenaar , de permissies , zijn grootte , en het tijdstip waarop de file het laatst gemodificeerd is. Deze informatie wordt gegeven in tabelvorm, gesorteerd op filenaam. Wat de permissies ("mode") betreft nog het volgende. Permissies worden weergegeven in de vorm van een serie tekens. De eerste kan zijn een "d" om aan te geven dat er sprake is van een directory, of een "-" voor een gewone file . Daarna komen negen tekens, waarvan de eerste drie de permissies voor de eigenaar van de file weergeven, de volgende drie die voor de groep (waartoe de eigenaar behoort) en de volgende drie die voor alle anderen . Voor elk type gebruiker worden de permissies dus met drie letters aangegeven. De eerste is "r" als de file door zo'n gebruiker gelezen mag worden, anders "-". Daarna komt een "w" voor een file die beschreven mag worden anders "-". Voor een directory geeft de derde letter, de "x", aan of dat directory doorzocht mag worden, zo niet "-". (Komt de "x" onder andere omstandigheden voor of staat er een "b", "c", "s" of "t" in de permissie-specificaties, dan zit er vermoedelijk iets fout, en gelieve u een der systeembeheerders te waarschuwen!). Wat de grootte ("size") betreft : hier staat het aantal symbolen, letters, cijfers, ed. waaruit de file bestaat. De eigenaar ("owner") : de "login-naam" van degene die de file aangemaakt heeft. Het modificatie tijdstip ("date") : het tijdstip, waarop de file voor het laatst is gewijzigd. Geven we het `ls` commando met de `-l` optie en een file-naam ipv. een directory-naam , dat is bv. `ls -l probleem1` , dan worden alleen de gegevens van die betreffende file verstrekt.

Voorbeeld :

```
$ ls -l uvxyz/noro
-rw-r----- 1 bigboss  37126 Apr  1 23:49 noro
$ ls -l uvxyz
-rwxrwxr-x 1 urgje      1225 Dec 25 20:01 exec
-rw-r----- 1 bigboss  37126 Apr  1 23:49 noro
drwxr-xr-- 2 urgje       32 Nov 11 11:11 subdir0
drwxrwxr-x 2 urgje      60 Dec 19 14:14 subdirl
$
```

U ziet dat "urgje" in directory "uvxyz" een file "exec" heeft die iedereen mag lezen, maar die alleen door "urgje" zelf en zijn groep beschreven mag worden, en die bovendien aan de systeem-beheerders gemeld moet worden vanwege het voorkomen van "x". Verder mag de file "noro" van "bigboss" alleen door "bigboss" zelf beschreven worden, terwijl er alleen leespermissie is voor hemzelf en de leden van zijn groep. Ook is te zien dat er in directory "uvxyz" twee subdirectories zijn. Daarvan is alleen subdirectory "subdir1" voor iedereen toegankelijk, hoewel gebruikers die niet in de groep van "urgje" zitten er niet in mogen schrijven. Het andere, "subdir0", is alleen voor "urgje" en de zijnen open.

De kolom met data en tijdstippen duidt op de creatie- cq. modificatie-gebeurtenissen. De kolom met de getallen "1225", "37126", "32" en "60" geeft voor files de omvang (in aantal karakters dat ze bevatten) en voor de subdirectories de grootte van de benodigde "administratie"-ruimte. En dan is er nog een kolom (vlak voor die van de eigenaars, hier bevattende de getallen 1 of 2) die u maar moet negeren omdat u zich toch niet bezig houdt met "linking".

- t sorteert niet op grond van de filenaam, maar op het modificatie tijdstip (de meest recent gemodificeerde file wordt het eerst genoemd).
- u neem ipv. van het modificatie tijdstip, het tijdstip waarop de file voor het laatst gebruikt ("use": gelezen of gemodificeerd) is.
- r draai de sorteervolgorde om, di. omgekeerd alfabetisch of oudste eerst (bij de -t optie).
- g geef de groepsnaam ipv. de naam van de eigenaar, di. geef aan tot welke groep de file behoort.

Voorbeeld: `ls -ltur` geeft de namen van alle files uit de huidige directory, inclusief de tijdstippen waarop ze het laatst gebruikt zijn. De meest recent gebruikte file wordt in dit geval het laatst genoemd.

ZIE OOK
chmod(4)

NAAM

`mkdir` - maak een directory

SYNOPSIS

`mkdir` directory ...

BESCHRIJVING

Mkdir creëert in de huidige directory een of meer subdirectories wier namen als "directory"-argumenten worden meegegeven. In aldus aangemaakte directories heeft iedereen leespermissie. Schrijfpermissie hebben alleen de eigenaar alsmede diegenen die tot dezelfde groep behoren. Wordt dit ongewenst gevonden dan moeten van zo'n directory de permissies veranderd worden (zie `chmod` (4)).

ZIE OOK

`chmod`(4), `ls`(4) en `rmdir`(4)

NAAM

mv - breng een file onder een andere naam (move it)
mv - zet een file over in een ander directory

SYNOPSIS

mv file1 file2
mv file directory

BESCHRIJVING

In het eerste geval wordt "file1" verwijderd, terwijl zijn inhoud wordt opgeslagen onder de naam "file2". In het tweede geval wordt "file1" ook verwijderd, maar zijn inhoud komt nu terecht in de aangegeven directory, weer onder de naam "file1" (er ontstaat "directory-naam/file1").

Voor een mv opdracht is altijd schrijfpermissie nodig op de directory waarin de hernoemde file terecht moet komen. Als "file2" (of "directory-naam/file1") al bestond voordat het mv commando uitgevoerd werd, raakt zijn inhoud verloren, doordat hij overschreven wordt. Het kan echter gebeuren, dat op "file2" (cq. "directory/file1") geen schrijfpermissie bestaat voor degene die het mv commando wil uitvoeren. Het mv commando antwoordt dan met "mode nnnn?", waarin "nnnn" een getal is, dat de permissies weergeeft (zie *chmod* (4)). Antwoordt de gebruiker met *y*, dan wordt het commando toch uitgevoerd. Het zonder toestemming overschrijven van files van een andere gebruiker lukt echter niet indien op de directory, waarin zo'n file zit, alleen die eigenaar zelf schrijfpermissie heeft.

ZIE OOK

cp(4)

PWD laat zien wat huidige directory is

1

NAAM

pwd - laat zien wat de huidige directory is (print working directory)

SYNOPSIS

pwd

BESCHRIJVING

zie onder NAAM .

ZIE OOK

chdir(4)

NAAM

rm - verwijder (remove) een file

SYNOPSIS

rm [opties] file ...

BESCHRIJVING

Rm verwijdert de files, die als argument genoemd worden. Verwijderen wordt gerekend als overschrijven. Indien de uitvoerder van *rm* zelf geen schrijfpermissie heeft, vraagt *rm* expliciet om toestemming. Deze vraag gaat in de vorm "mode nnnn ?", waarbij "nnnn" een getal is, dat de permissies aangeeft (zie *chmod* (4)). Als met *y* geantwoord wordt, wordt de verwijdering alsnog uitgevoerd. Verwijderen vereist echter hoe dan ook, dat degene die een file verwijderen wil, schrijfpermissie in de betreffende directory heeft. Op die manier kunt u dus verhinderen dat anderen uw files weggooien.

Opties:

- r Als de file geen gewone file maar een directory is, wordt deze te samen met alle files die zich in die directory (of subdirectories) bevinden, verwijderd. PAS HIERMEE DUS OP!
- i De in de argumenten aangegeven files worden alleen verwijderd, nadat het commando nog eens expliciet om toestemming heeft gevraagd. (antwoord met *y* or *n*). Deze optie is vooral handig, samen met de -r optie: type *rm -ri directorynaam* , en van al de files uit uw huidige directory en zijn subdirectories, wordt dan één voor één gevraagd of ze weggegooid mogen worden. U kunt met *y* antwoorden als dit inderdaad het geval is (zo niet : *n*).

ZIE OOK

mv(4)

NAAM

rm~~dir~~ - verwijder (remove) een directory

SYNOPSIS

rm~~dir~~ directory ...

BESCHRIJVING

Rmdir verwijdert de als argument(en) genoemde directories, doch alleen als ze leeg zijn, d.i. geen files of subdirectories bevatten (zie *rm* (4)). Verder is vereist dat diegene die een directory verwijderen wil, schrijffpermissie heeft in de een niveau hoger gelegen directory, waarvan de te verwijderen directory een subdirectory is.

ZIE OOK

chmod(4), mkdir(4) en rm(4)

"this page is left blank intentionally"

HOOFDSTUK 5 DIVERSEN

Inleiding

Hierin komen enkele moeilijk te klassificeren maar wel nuttige commando's aan de orde. En dat zijn dan :

<i>banner</i>	<i>passwd</i>
<i>cal</i>	<i>sara</i>
<i>calendar</i>	<i>setff</i>
<i>crypt</i>	<i>sort</i>
<i>date</i>	<i>split</i>
<i>login</i>	<i>tail</i>
<i>mail</i>	<i>who</i>
<i>mesg</i>	<i>write</i>

Met behulp van het commando *login* verkrijgt u toegang tot het systeem. En met het commando *passwd* beveiligt u het systeem tegen het ongeoorloofd gebruik van uw login-naam door anderen en voorkomt u tevens dat zulke anderen mbv. uw "eigenaar-permissies" (zie uitleg bij bv. *ls* (4) en *chmod* (4)) aan uw eigen files gaan knoeien.

Dank zij het commando *who* is het mogelijk vast te stellen welke gebruikers er nog meer "ingelogd" zijn op het systeem. Met *write* kunt u dan met deze gebruikers "converseren". Het commando *mesg* is er dan om binnenkomende "conversatie" te weren op momenten dat uw terminal-werk geen storingen kan velen. Verder kunt u met alle gebruikers (ook die niet "ingelogd" zijn) "corresponderen" mbv. *mail*.

Met *date* krijgt u de datum en het tijdstip van de dag, terwijl u met *cal* een kalender krijgt van een door u gekozen maand of jaar. Verrassend is voor u ongetwijfeld, dat *calendar* geen kalender levert, maar u in staat stelt een agenda bij te houden.

Met *tail* kunt u de "staart" van een file apart bekijken en met het commando *split* kunt u een file in kleinere stukken splitsen.

Het commando *sort* is er, zoals u zou verwachten, om u in staat te stellen in uw gegevens-bestanden sorteerwerk te verrichten.

En om het u mogelijk te maken uw gegevens voor ieder ander geheim te houden is in deze verzameling het commando *crypt* aanwezig.

Dan is er ook nog het commando *sara*. Hiermee kunt files uitwisselen tussen ons PDP 11/34 -systeem en het CDC/CYBER -systeem van SARA.

Van een geheel andere soort is dan weer het *setff* commando, waarmee u controle krijgt over het pagineren tijdens het afdrukken van een file op een HCT-printer.

Tenslotte is er nog een commando waarvan het nut voor uw werk geheel duister is, nl. *banner*. Maar misschien weet u toch nog wel een toepassing te vinden voor de door *banner* geleverde grote, samengestelde letters.

NAAM

banner - druk tekst af in grote, samengestelde letters

SYNOPSIS

banner woord ...

BESCHRIJVING

Ipv. een beschrijving volgt hier een voorbeeld van gebruik :

banner monster voor U

levert

```
*   *   ****   *   *   ****   *****   *****   *****
**  **  *   *  **  *   *   *   *   *   *   *   *   *   *
* ** *  *   *  * *  *   ****   *   *****   *   *
*   *  *   *  * *  *   *   *   *   *   *   *   *****
*   *  *   *  * *  **  *   *   *   *   *   *   *   *
*   *   ****   *   *   ****   *   *****   *   *
```

```
*   *   ****   ****   *****
*   *  *   *  *   *  *   *   *
*   *  *   *  *   *  *   *   *
*   *  *   *  *   *  *   *****
*  *  *   *  *   *  *   *   *
**   ****   ****   *   *
```

```
*   *
*   *
*   *
*   *
*   *
```

en hopelijk is dat duidelijk genoeg.

NAAM

cal - lever kalender

SYNOPSIS

cal [maandnr] jaartal

BESCHRIJVING

cal print op uw terminal de kalender voor het aangegeven jaar af. Indien ook de maand opgegeven is, dan beperkt *cal* zich tot die maand. Het jaartal kan gekozen worden uit de getallen 1 tot en met 9999. De maand dient te worden opgegeven als een nummer: 1,2,3, ... , of 12 . Denkt u er overigens even aan dat *cal 85* u niet de kalender van 1985 levert, maar die van een negentien-honderd jaar eerder.

Verder dient u zich bij jaartallen uit het verleden te realiseren dat dit *cal* -programma geschreven is voor engelstalige gebieden. Dit houdt in dat vóór 14 september 1752 uitgegaan wordt van de Juliaanse tijdrekening ipv. de Gregoriaanse. Bovendien laat *cal* ieder jaar met de maand januari beginnen, hetgeen voor een groot aantal jaren in het begin van de jaartelling niet juist is.

ZIE OOK

date(5)

NAAM

calendar - geheugensteun / agenda-service

SYNOPSIS

calendar [-]

BESCHRIJVING

Calendar gaat na of er een file genaamd "calendar" bestaat in het huidige directory en gaat, indien dat het geval is, de tekst van deze file onderzoeken op de aanwezigheid van de geldende datum en/of de datum van de volgende (werk)dag. Die regels die deze data bevatten worden uitgeprint. Alleen data die passen in de engelse notatie, zoals "Jan. 5" of "january 6" of "12/31", worden herkend in de tekst. Niet herkend worden dus : "8 January" of "31/12". Het zal duidelijk zijn dat de genoemde file "calendar" door de gebruiker zelf op een zinvolle wijze "up to date" moet worden gehouden.

Indien aan *calendar* een willekeurig argument meegegeven wordt, dan is het werkings-terrein van *calendar* : alle gebruikers die een file "calendar" in hun "login directory" hebben. De tekstregels die de gezochte dateringen bevatten worden echter niet rechtstreeks uitgeprint, maar via *mail* (5) opgeborgen in de "brievenbus" ("mbox") van de betreffende gebruiker. Omdat het gebruik van deze optie mogelijk tot vertragingen in het systeem leidt, wordt men verzocht slechts in overleg met het systeem-beheer hiervan gebruik te maken.

ZIE OOK

date(5) en mail(5)

NAAM

`crypt` - vercijfer of ontcijfer

SYNOPSIS

`crypt` [*sleutel*] <*infile*> *uitfile*

BESCHRIJVING

Het commando *crypt* wordt gebruikt voor het vercijferen (cryptograferen) en ontcijferen van teksten. Het dient alleen gebruikt te worden, als de meest verregaande beveiliging (geheimhouding) van de tekst noodzakelijk is.

Als "*infile*" een file met gewoon leesbare tekst is, leidt het kommando *crypt sleutel <infile> uitfile* tot het aanmaken van een vercijferde tekst op "*uitfile*" (let op het gebruik van "<" en ">"). Wordt daarna "*infile*" weggegooid (zie *rm* (4)) dan is de tekst daarna niet meer in een leesbare vorm in het systeem aanwezig.

Het wederom ontcijferen van de tekst op "*uitfile*" gaat met: *crypt sleutel <uitfile> uitfile1*. De weer ontcijferde, en dus leesbare tekst komt hier op "*uitfile1*". Het vercijferen (cq. ontcijferen) gebeurt op een manier die bepaald wordt door het "*sleutel*"-argument. Hiervoor is de gebruiker vrij een willekeurig woord van maximaal elf tekens te kiezen. In het algemeen zal een sleutel van drie á vier tekens voldoende zijn. Indien het "*sleutel*"-argument niet meegegeven wordt in de commandoregel, dan zal *crypt* er om vragen door "Enter key:" op uw terminal af te drukken. De sleutel die u daarachter intypt wordt dan door *crypt* onzichtbaar gehouden, zoals u dat kent van *passwd* (5). En eigenlijk is dit een veel betere manier om *crypt* te gebruiken.

Opmerking: de sleutel waarmee een tekst vercijferd is, is absoluut noodzakelijk om deze weer te kunnen ontcijferen, want een vergeten sleutel betekent een onherroepelijk verloren tekst .

Nog een voorbeeld:

```
crypt p32 <begroting> gehbeg
rm begroting
crypt p32 <gehbeg> begroting
```

Met sleutel "p32" wordt de file "begroting" vercijferd, het resultaat komt op "gehbeg". Daarna wordt "begroting" verwijderd, er is nu dus geen leesbare versie meer van. Tenslotte wordt "gehbeg" weer ontcijferd waarbij wederom een leesbare "begroting" ontstaat.

DATE geef de datum

1

NAAM

date - geef de datum

SYNOPSIS

date

BESCHRIJVING

Date vertelt u welke dag het is, en hoe laat dan wel.

ZIE OOK

cal(5), calendar(5)

NAAM

login - systeemingang

SYNOPSIS**login** [loginnaam]BESCHRIJVING

Het commando *login* wordt gebruikt om toegang te krijgen tot het computer-systeem. Indien *login* aangeroepen wordt zonder argument (en dat is automatisch zo indien u de vraag "MC/CLASS CODE: " van het *MICOM-SYSTEEM* - dat het netwerk van MC-computers en hun terminals bestuurt - met 3, 4, 5 of 6 hebt beantwoord), dan wordt er achtereenvolgens naar uw gebruikersnaam (login-naam) en het bijbehorende "password" gevraagd. Het programma *login* zal na de vraag om het "password" het type-mechanisme ontkoppelen van uw beeldscherm (dan wel terminal-papier) totdat u weer een *RETURN* aanslaat. De bedoeling van deze ontkoppeling is natuurlijk de geheimhouding van uw "password" te bevorderen door het onzichtbaar voor anderen te houden.

Indien u reeds "ingelogd" bent, kunt mbv. het commando *login* onder een andere gebruikersnaam het systeem binnenkomen. De *CNTL d*, die anders voor uitloggen nodig is, kan dan achterwege blijven omdat *login* daar nu voor zorgt. In zulk een situatie is dan ook de toevoeging van het "loginnaam"-argument zinvol, bv. *login play*.

Het commando *login* zorgt er ook voor dat u attent gemaakt wordt op de aanwezigheid van eventuele post, u toegezonden middels *mail* (5).

ZIE OOK

mail(5) en passwd(5)

NAAM

mail - verzenden en ontvangen van gebruikers-post

SYNOPSIS

mail gebruiker ...
mail [-r] [-q] [-p] [-f file]

BESCHRIJVING

Het commando *mail* is een communicatie-middel tussen gebruikers van het systeem zonder dat, zoals bij *write* (5), zendende en ontvangende gebruiker tegelijkertijd "ingelogd" dienen te zijn. Elke gebruiker krijgt, zodra hij weer "inlogt", een melding in het geval dat er in zijn "brievenbus" zulke binnengekomen correspondentie ligt te wachten. De "post" kan uit meerdere berichten bestaan; elk daarvan is voorzien van de naam van de afzender en een "postmerk". De regels waarop deze "postmerken" zijn geplaatst, beginnen met een ">".

Mail, zonder argumenten, drukt op de terminal de binnengekomen en/of nog bewaarde "post" af. Dit gebeurt dan bericht voor bericht, en wel in de omgekeerde volgorde van binnenkomst.

Na ieder afgedrukt bericht vraagt *mail* de gebruiker, middels een "?", een aanwijzing betreffende de verdere afhandeling van dat bericht. U kunt als aanwijzing een van de volgende antwoorden intypen:

RETURN ga verder naar volgende bericht;

d verwijder het bericht uit het systeem en ga verder met het volgende bericht;

p druk het bericht nog een keer af;

- ga terug en druk het vorige bericht nog een keer af;

s [F] bewaar dit bericht en schrijf het daartoe op een file met de naam die in de string "F" staat ("mbox" als er geen "F" meegegeven is);

w [F] bewaar dit bericht, zonder de koptekst, en schrijf het daartoe op een file met de naam die in de string "F" staat (standaard is "mbox");

m [G] stuur het bericht naar systeem-gebruiker "G" (bij weglating van de gebruikers-naam wordt het bericht naar uzelf doorgestuurd);

CNTL d (of *EOT*, di. end-of-text) stop de nog niet behandelde berichten terug in de "brievenbus" ("mbox") en stop;

q "quit", dwz. ophouden, dwz. doe als bij *CNTL d*;

x stop, en wel zonder de inhoud van de "brievenbus" te veranderen;

- !c wacht even met de post-behandeling zodat de systeem-commando's, die in de string "c" staan, uitgevoerd kunnen worden, ga verder met de post-behandeling zodra het systeem met een "!" geantwoord heeft;
- ? druk eerst even een lijst af *mail* -aanwijzingen, want ik ben ze vergeten;

Bij het commando *mail* kan men de volgende opties meegeven :

- r afhandeling van de binnengekomen berichten geschiedt nu in volgorde van binnenkomst;
- q een door de gebruiker gegeven "interrupt" (di. een *DEL* , *DELETE* of *RUBOUT*), waardoor het afdrukken van een bericht tussentijds wordt beëindigd, heeft nu niet een eventueel neveneffect op de inhoud van de "brievenbus";
- p de vraagstelling "?" na ieder bericht wordt onderdrukt, zodat alle post achter elkaar wordt afgedrukt;
- f de hierbij genoemde file (en als er geen file-naam meegegeven wordt, neemt *mail* hiervoor "mbox") wordt afgedrukt als "brievenbus".

Wanneer één of meer gebruiker's-namen genoemd zijn in het "gebruikers"-argument van de commando-regel, dan leest *mail* alles wat erna ingetypt wordt in. Dit inlezen wordt door de gebruiker gestopt middels een regel met enkel een *CNTL d* of enkel een *punt* . En dat is dan tevens een startsein voor *mail* om het ingetypte als "post" over te brengen naar de vermelde gebruikers. U kunt echter ook eerst uw bericht met de *editor* samenstellen en op een file, bv. "briefje" , schrijven. Het voordeel daarvan is de mogelijkheid tot het aanbrengen van correcties in de tekst. Uw tekst komt dan bij gebruiker "lastpost" door het geven van het commando *mail lastpost < briefje* .

De mee te geven "gebruikers-namen" zijn de door het computer-systeem gebruikte "login-namen" .

ZIE OOK

login(5) en write(5)

NAAM

mesg - tolereer of verbiedt inkomende *write* -berichten

SYNOPSIS

mesg [*n*] [*y*]

BESCHRIJVING

Mesg met als argument *n* verbiedt het toezenden door andere gebruikers van *write* -berichten (zie *write* (5)) naar uw terminal. Een commando *mesg y* herstelt de oude situatie weer, terwijl *mesg* zonder een argument de *mesg* -status van dat moment zal rapporteren.

ZIE OOK

write(5)

NAAM

passwd - zet of verander het "password"

SYNOPSIS

passwd

BESCHRIJVING

Passwd moet aangeroepen worden, als u uw "password" wilt instellen of veranderen.

Zo u al een "passwd" had, vraagt het programma eerst om uw oude "password" om te kijken "of u het echt wel zelf bent". Daarna wordt twee keer achter elkaar om het gewenste nieuwe "password" gevraagd (twee keer om tikfouten uit te sluiten). Een opgegeven "password" moet minsten vier tekens lang zijn als er voldoende variatie in de tekens zit (hoofd- en kleine letters door elkaar bijvoorbeeld) en minsten zes tekens als één lettertype (bijvoorbeeld de kleine letter) gebruikt wordt. Kortere "passwords" worden geweigerd.

ZIE OOK

login(5)

NAAM

sara - verstuur of ontvang (via SARA) files

SYNOPSIS

sara [opties] [nummer]

BESCHRIJVING

Sara wordt aangeroepen om files te versturen naar de CDC/CYBER computer waarover de Stichting Academisch Rekencentrum Amsterdam beschikt. Ook kunnen files van SARA ontvangen worden. U ontvangt mail indien de file verwerkt is. Er wordt altijd een copie van de file verzonden, zodat uw oude file beschikbaar blijft.

Zonder opties werkt het commando als volgt (onderstreept schrift duidt aan wat de computer antwoordt):

% sara	/zonder opties
<u>type command</u>	
<u>To</u> , <u>Copy to</u> , <u>From</u> , <u>Lineprinter</u> of <u>Punch op kaarten</u>	
<u>to</u>	/of simpel t
<u>type filenaam</u>	/uw huidige filenaam
<u>type SARA permanent filenaam</u>	/alles wat van SARA hier mag
<u>file naam</u>	/staan, dus ook cy= xr= oid.
<u>type code</u>	/sara hanteert 2 soorten codes
<u>display(standaard) of ascii95</u>	/niets intypen betekent d
<u>display</u>	/of simpel d

Merk op: De USERID bij SARA ligt vast en staat op ID=MCPONS, eventueel kan de -f vlag worden meegegeven om dit te voorkomen.

Na deze handelingen krijgt u een nummer geretourneerd, dat gebruikt kan worden voor het opvragen van de status van uw job. Opvragen gaat (voor bv. jobnr. 501) dan als volgt:

% sara 501	
<u>sara naam</u> :	file naam
<u>file naam</u> :	file
<u>command</u> :	to
<u>code</u> :	Display
<u>jobnr</u> :	501
<u>inputnr</u> :	1
	/dit is uw eigenlijke nummer

Indien u geen mail hebt ontvangen en uw job is ook niet te op te vragen met " sara 501 " dan wordt uw file op dat moment juist overgezonden. Een enkele keer kan het ook betekenen dat uw opdracht is verloren gegaan; waarschuw dan een der systeembeheerders.

Opties bij het sara commando zijn:

- a vraag de (verkorte) status op van alle jobs die zich in de input-queue bevonden;
- m er wordt U geen mail gezonden van het resultaat (niet aan te raden);
- s *COPYSBF* effect, hetgeen een term is voor SARA-kenners. De computer bij SARA gebruikt nl. het eerste karakter per regel als printer-control-karakter, bv. "lwillekeurige tekst" betekent "FFwillekeurige tekst". Dit is vooral van belang bij het ontvangen van files, indien niet iedere regel op de SARA permanent file begint met een *SPATIE*, dan gaat het eerste karakter verloren of er komt iets te staan wat u niet wilt.
Bij het versturen van files naar SARA heeft -s dan tot gevolg dat uw file vóór het versturen aangepast wordt, waardoor *BACKSPACES* ed. "overprints" tot gevolg hebben (di. een + in de eerste kolom bij SARA). Voor normaal gebruik is deze optie nodeloos tijdverslindend.
- f deze vlag wordt meegegeven indien u uw eigen accountnummer informatie wilt meegegeven, het is dan ook mogelijk op eigen ID te catalogiseren.

STAGNATIES

Deze mogen natuurlijk niet optreden. Er wordt tegen beschermd en mocht SARA "down" gaan, dan wordt op een later tijdstip (waar u bericht van krijgt) getracht de job opnieuw te versturen.

Het heeft (nog) geen zin gecryptografeerde files te versturen, wel bent u zoveel mogelijke beschermd tegen het lezen van uw accountnummer en password informatie.

SETFF stel de *FORMFEED* in op een HCT printer

1

NAAM

setff - stel op een HCT de *FORMFEED* in

SYNOPSIS

setff

BESCHRIJVING

Door middel van de aanroep *setff* wordt uw HCT-printer geschikt gemaakt om bij ontvangst van een *FORMFEED* karakter (*FF*) daadwerkelijk op een nieuwe pagina te verder te gaan.

De aanroep heeft allereerst een neveneffect, waarbij een groot stuk papier-opvoer plaats vindt. Indien ook u dit te verkwistend vindt, kunt dit te niet doen door tijdens dit papier-transport de rechter grote zwarte knop van de rol naar links ingedrukt te houden.

Daarna draait u het papier tot precies boven aan de pagina en tijdens de rest van de sessie zal de HCT u inderdaad een nieuwe pagina geven zodra daartoe de opdracht ontvangen wordt.

De fijnproevers verwijzen we naar het aparte HCT-manual, dat te verkrijgen is bij de systeembeheerders. *Setff* produceert achtereenvolgens:

ESC 6 FF ESC pp ESC 5 FF

NAAM

sort - sorteer, of voeg reeds gesorteerde files samen

SYNOPSIS

sort [opties] [+pos1] [-pos2] ... [-o outfile] infile ...

BESCHRIJVING

Sort kan zowel gebruikt worden om regels van een of meer files te sorteren als om reeds gesorteerde files te mengen.

De namen van de te sorteren (te mengen) files worden geplaatst in het "infile ..." -argument. Het resultaat gaat naar de terminal, tenzij het "-o outfile" -argument gebruikt wordt (de uitvoer van *sort* komt dan op de in "outfile" meegegeven file terecht).

Worden geen andere argumenten dan de namen van invoer- en uitvoerfiles meegegeven, dan sorteert *sort* op grond van de volledige invoerregels; elk symbool uit elke regel telt mee.

De "pos" -argumenten dienen om de sorteersleutel te beperken tot delen van de regels (zoals het "n-de woord"), de andere (één letterige) argumenten bepalen oa. welke symbolen uit de sleutel meedoen met sorteren, en wat de rangorde van deze symbolen is.

-i De symbolen op grond waarvan gesorteerd wordt zijn (in volgorde van hun rangorde):

SPATIE ! " # \$ % & ^ () + , - . 0 t/m 9 : ;
< = > ? @ A t/m Z [\] ^ _ ` a t/m z { | } ~

Zonder de -i optie zouden ook TABs, NEWPAGE (FF), RETURN (CR), BACKSPACE etc. meedoen met sorteren.

Voorbeeld: "medewerk(st)er" komt vóór "medewerker", want "(" komt vóór "e".

-d Gebruik alleen letters, cijfers en SPATIES bij het sorteren. Leestekens ed. spelen geen rol.

Voorbeeld: "medewerker" komt vóór "medewerk(st)er".

-n Nuttig als de sorteersleutel (regel) begint met een getal bestaande uit nul of meer cijfers, eventuele SPATIES, een eventuele decimale punt en eventueel een puntteken. Sorteren vindt dan plaats op grond van numerieke waarde (laagste eerst).

Opmerking: Meestal zal de -d optie gewenst zijn, vermoedelijk zal in de overgrote meerderheid der gevallen minstens -i, -d of -n gebruikt moeten worden.

-b verwaarloos eventuele SPATIES of TABs aan het begin van een sorteersleutel;

-f behandel hoofdletters en corresponderende kleine letters op dezelfde manier, di. "de", "De", "DE" en "dE" zijn wat het sorteren betreft, identiek (zonder -f is de volgorde "DE", "De", "dE", "de");

- r Draai de sorteervolgorde om, bv. -dr geeft omgekeerd alfabetische volgorde, -nr sorteert op een numerieke sleutel, van groot naar klein.

Enige voorbeelden:

Stel "woordwoud" is een woordenlijst, met één woord per regel.

Dan plaatst

sort -i woordwoud -o letterpark

een gesorteerde woordenlijst op de file "letterpark". En

sort -d woordwoud -o letterpark

doet hetzelfde, maar kijkt alleen naar letters en cijfers. Als elk woord in "woordwoud" voorafgegaan zou worden door een of ander nummer, dan geeft

sort -nd woordwoud -o letterpark

een sortering op nummer; regels die met hetzelfde nummer beginnen worden alfabetisch geordend.

- u Van identieke regels (of regels met identieke sorteersleutels) wordt er telkens maar één in de uitvoer geplaatst. "Identiek" heeft hier betrekking op de symbolen die bij het sorteren meedoen; onder de -d optie is "medewerk(st)er" identiek aan "medewerkster".
- m voeg twee of meer reeds gesorteerde invoerfiles samen tot één nieuwe gesorteerde file;
- c controleer of de invoerfile gesorteerd is; zo ja geef dan géén uitvoer, zo nee, sorteer en stuur uitvoer naar de terminal of uitvoerfile.
- tC verander de standaard veldscheider (de SPATIE) zodat deze veldscheider nu het in "C" meegegeven symbool is (voor het begrip "veldscheider" zie hieronder bij sorteersleutels & "pos" -argumenten).

Nu het vaststellen van sorteersleutels en de "pos" -argumenten. De regels van de invoerfiles zijn opgebouwd uit velden, standaard is een veld een serie symbolen, begrensd door SPATIES (of het begin of eind van een regel). Sorteersleutels worden nu aangegeven door vast te stellen, uit welke velden ze bestaan. Per sorteersleutel komt een argument vd. vorm *+pos1 -pos2* (of alleen *+pos1*) voor.

Voorbeelden:

- +2-4 De sorteersleutel bestaat uit het 2e en 3e veld, di. het 2e tot het 4e veld.
- +2-4,+8-11 Sorteert eerst op het 2e en het 3e veld, regels die daarbij op dezelfde plaats komen, worden verder gesorteerd op grond van hun 8e, 9e en 10e veld.
- +3-7,+11 Sorteert op veld 3 tot 7 (di. 3 t/m 6), daarbinnen op de velden vanaf het 11e t/m het laatste.

Het is ook mogelijk sorteersleutels te laten beginnen of ophouden binnen velden bv.

+3.2-7.6 De sleutel begint met het 2e symbool van het 3e veld en loopt door tot en met de eerste 5 symbolen van het 7e veld.

Bij het opgeven van sorteersleutels kan gebruik gemaakt worden van letters *b*, *d*, *f*, *i*, *n*, *r* waarvan de betekenis hierboven al besproken is. Voorbeelden:

+3df-7 De sleutel bestaat uit de velden 3 t/m 6, sorteren gebeurt op grond van letters en cijfers, er wordt geen onderscheid gemaakt tussen hoofd- en kleine letters.

sort -dr +3n-7 +8-12 +15 -o druit drin:

De inhoud van de file "drin" wordt gesorteerd, het resultaat gaat naar "druit". Er zijn hierbij drie sorteersleutels: veld 3 t/m 6, veld 8 t/m 12 en veld 15 en volgende. Sorteren geschiedt op grond van letters en cijfers, in omgekeerd alfabetische volgorde (*-d* optie), behalve voor de eerste sleutel, waarvoor numerieke volgorde gespecificeerd is. Deze numerieke ordeningen van klein naar groot, di: de globale *-r* optie heeft hier geen effect. Er geldt nl. de volgende regel:
Globaal (buiten een specificatie sleutel) gespecificeerde opties gelden voor alle sleutels waarin wel opties gespecificeerd worden. Voor sleutels waarin wel opties gespecificeerd worden, gelden deze opties en verder de standaard situatie.

NAAM

`split` - splits een file

SYNOPSIS

`split` [`-n`] file [copienaam]

BESCHRIJVING

Door het commando *split* wordt de file, met bv. de naam "filnam", gekopieerd, en wel in stukken van zoveel regels als in de "`-n`" -optie wordt meegegeven. Bij weglating van deze optie hanteert *split* een aantal van 1000 regels. De verkregen copie-stukken zullen als naam krijgen de string vermeld in "copienaam" met daarachter vastgeplakt achtereenvolgens de toevoegsels (naar gelang nodig) "aa", "ab", , "az", "ba", "bb", etc . Indien er geen "copienaam" in de commando-regel is opgenomen, wordt door *split* daarvoor de letter "x" gebruikt .

NAAM

tail - geef het staartje van een file

SYNOPSIS

tail [+N[opties]] file

BESCHRIJVING

Tail copieëert de genoemde file vanaf een zekere plaats naar uw scherm. Die zekere plaats wordt bepaald door het getal dat in "N" wordt meegegeven, waarbij dan nog + een keuze voor de gebruiker biedt: een "+" voor telling vanaf het begin en een "-" als de telling vanaf het einde van de file gerekend moet worden. Via de opties geeft men dan aan wat er geteld moet worden :

-l de telling geschiedt in regels ;

-c de telling geschiedt in karakters ;

Bij eventueel ontbreken van opties geschiedt de telling in regels.

Dit commando kan van pas komen bij grote, door programma's (bv. *formatter*) geproduceerde files. Eventuele foutmeldingen betreffende de executie van het producerende programma staan vaak (ook) aan het eind van de uitvoer vermeld, en als die uitvoer-file groot is kan het prettiger zijn eerst even alleen naar de staart ervan te kijken.

NAAM

who - wie is er op het systeem

SYNOPSIS

who [am I]

BESCHRIJVING

Who , zonder argument, geeft per gebruiker, die op dat moment "ingelogd" is: de "login-naam", de "terminal-naam", en het tijdstip van "inloggen". Met de "terminal-naam" geeft *who* aan op welke wijze de betreffende terminal door het computer-systeem aangeduidt wordt (en welke aanduiding voor u van belang kan zijn bij het gebruiken van *write* (5)).

Bij het gebruik van argumenten, zoals in *who am I* of *who are you* , heeft de geleverde informatie betrekking op alleen dié gebruiker die "ingelogd" is op de terminal, waarop het commando werd ingetypt. Deze versie is bijvoorbeeld nuttig bij het aantreffen van een ogenschijnlijk verlaten terminal : er is iemand op "ingelogd" die in velden noch wegen te zien is, en via *who* kunt u nu te weten komen aan wie nu eigenlijk moet vragen of hij/zij instemt met "uitloggen".

ZIE OOK

login(5) en *write*(5)

NAAM

write - schrijf naar een andere gebruiker die "ingelogd" is

SYNOPSIS

write gebruiker [ttynr]

BESCHRIJVING

Write dient, om naar een andere gebruiker te schrijven, mits die op dat moment is "ingelogd" (gebruik *who* (5) om daar achter te komen).

Indien echter de ontvangende gebruiker meer dan één maal onder dezelfde "login-naam" "ingelogd" is, ontstaat er voor *write* een dubbelzinnige situatie. Het "ttynr"-argument toevoegen verhelpt het probleem: u typt dan op die plaats de letterlijke tekst van de terminal-aanduiding in. Deze terminal-aanduiding vindt u door *who* (5) aan te roepen en in de tweede kolom te kijken op die regels waar de door u gekozen gebruiker staat vermeld.

Voorbeelden: *write user1 tty13* of *write user2* .

Nadat u het *write* commando gegeven hebt, verschijnt alles wat u verder intypt op de terminal van de andere gebruiker. Om deze toestand te beëindigen typt u *CNTL d* in. Op de terminal van de andere gebruiker verschijnt dan EOF, ter aanduiding dat u de communicatie beëindigt heeft.

Wilt u communicatie met een andere gebruiker in "gespreksvorm" onderhouden, dan zal deze terug moeten kunnen schrijven. Om te voorkomen dat vraag en antwoord door elkaar heenlopen, dient u elke mededeling af te sluiten dmv. van een "0" voor "over", ingetypt aan het begin van een regel. Wacht daarna op antwoord van uw gesprekspartner, ook hij moet zijn antwoord beëindigen met "0". Degene die het gesprek wil beëindigen kan dan ook in zo'n situatie het beste, overeenkomstig de "0", eerst "0&0" ("over and out") intypen alvorens *CNTL d* te geven.

Wilt u niet gestoord worden door boodschappen van andere gebruikers, dan kunt u dat voorkomen mbv. *mesg* (5) .

Het sturen van een bericht naar een gebruiker, welke niet is "ingelogd", kan via het *mail* (5) commando.

ZIE OOK

mail(5), mesg(5) en who(5)

UNIX ASCII karakter code :

	0	1	2	3	4	5	6	7
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL
8	BS	HT	NL	VT	NP	CR	SO	SI
16	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB
24	CAN	EM	SUB	ESC	FS	GS	RS	US
32	SP	!	"	#	\$	%	&	'
40	(	)	*	+	,	-	.	/
48	0	1	2	3	4	5	6	7
56	8	9	:	;	<	=	>	?
64	@	A	B	C	D	E	F	G
72	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W
88	X	Y	Z	[	\	]	^	_
96	`	a	b	c	d	e	f	g
104	h	i	j	k	l	m	n	o
112	p	q	r	s	t	u	v	w
120	x	y	z	{		}	~	DEL

besturings-karakters :

(druk de CNTL-toets in tesamen met de tussen haakjes aangegeven toets)

(`)	NUL	null	(p)	DLE	data link escape
(a)	SOH	start of heading	(q)	DC1	device control 1
(b)	STX	start of text	(r)	DC2	device control 2
(c)	ETX	end of text	(s)	DC3	device control 3
(d)	EOT	end of transmission	(t)	DC4	device control 4
(e)	ENQ	enquiry	(u)	NAK	negative acknowledge
(f)	ACK	acknowledge	(v)	SYN	synchronous idle
(g)	BEL	bell	(w)	ETB	end of transmission block
(h)	BS	backspace	(x)	CAN	cancel
(i)	HT	horizontal tab	(y)	EM	end of medium
(j)	NL	new line	(z)	SUB	substitute
(k)	VT	vertical tab	([	ESC	escape
(l)	NP	new page (formfeed)	(\)	FS	file separator
(m)	CR	carriage return	(])	GS	group separator
(n)	SO	shift out	(^)	RS	record separator
(o)	SI	shift in	(_)	US	unit separator
			(-)	DEL	delete / rubout

grafische karakters :

SP	space	+	plus sign	@	(commercial) at symbol
!	exclamation point	,	comma	[	left (square) bracket
"	quotation mark	-	hyphen /minus sign	\	backslash
#	number sign / sharp / pound symbol	.	period	]	right (square) bracket
\$	currency symbol / dollar symbol	/	slant / slash	^	circumflex / caret
%	percent sign	:	colon	_	underlining / underscore
&	ampersand	;	semicolon	`	grave accent
'	apostrophe	<	less than symbol	{	opening brace / left (curly) bracket
(	opening parenthesis	=	equals symbol		(broken) vertical bar
)	closing parenthesis	>	greater than symbol	}	closing brace / right (curly) bracket
*	asterisk	?	question mark	~	tilde

geef het	aantal regels in een file	<i>wc</i> (1)
geef het	aantal woorden in een file	<i>wc</i> (1)
geheugensteun /	agenda-service	<i>calendar</i> (5)
	blader file door	<i>more</i> (1)
copiëer naar	cassette	<i>wrcas</i> (2)
copiëer	cassette naar file	<i>rdcas</i> (2)
maak pons-	concept aan	<i>concept</i> (3)
"	converseer" met andere gebruikers	<i>write</i> (5)
	copiëer file	<i>cp</i> (2)
	copiëer files naar cassette	<i>wrcas</i> (2)
	copiëer cassette naar file	<i>rdcas</i> (2)
"	correspondentie" tussen gebruikers	<i>mail</i> (5)
	data-entry editor	<i>ded</i> (3)
geef	datum	<i>date</i> (5)
ga naar ander	directory	<i>cd</i> (4)
ga naar ander	directory	<i>chdir</i> (4)
verwijder	directory	<i>rmdir</i> (4)
maak nieuw	directory	<i>mkdir</i> (4)
toon wat huidige	directory is	<i>pwd</i> (4)
geef inhoud van	directory weer	<i>ls</i> (4)
file	doorbladeren	<i>more</i> (1)
zoek spelfouten in	engelse tekst	<i>spell</i> (1)
data-entry	editor	<i>ded</i> (3)
"	edit" tekstinvoer	<i>editor</i> (3)
verander	file-naam	<i>mv</i> (4)
stel op HCT	<i>FORMFEED</i> in	<i>setff</i> (5)
verzend/ontvang	gebruikers-post	<i>mail</i> (5)
agenda-service /	geheugensteun	<i>calendar</i> (5)
samenvoegen van	gesorteerde files	<i>sort</i> (5)
print	grote, samengestelde letters	<i>banner</i> (5)
stel op	HCT <i>FORMFEED</i> in	<i>setff</i> (5)
kwaliteits print op	HCT terminal	<i>hqp</i> (2)
kwaliteits print op	HYPRINT terminal	<i>hqp</i> (2)
tolereer/verbied	inkomende write-berichten	<i>mesg</i> (5)
lever	kalender	<i>cal</i> (1)
verander	lees- of schrijfpermissie	<i>chmod</i> (4)
samenvoegen van	ongesorteerde files	<i>cat</i> (1)
vercijfer of	ontcijfer file	<i>crypt</i> (5)
	ontvang gebruikers-post	<i>mail</i> (5)
zet/verander "	password"	<i>passwd</i> (5)
verander	permissie(s)	<i>chmod</i> (4)
maak	ponsconcept aan	<i>concept</i> (3)
	print file	<i>pr</i> (2)
	print file	<i>print</i> (2)
	print file	<i>pt</i> (2)
	print grote, samengestelde letters	<i>banner</i> (5)
	print file op de QUME printer	<i>qupr</i> (2)
	print file(s) op terminal	<i>cat</i> (1)
kwaliteits	print op HCT terminal	<i>hqp</i> (2)
kwaliteits	print op HYPRINT terminal	<i>hqp</i> (2)
	print op LA-120 matrix printer	<i>lpr</i> (2)
zoek bepaalde	regels in een file	<i>grep</i> (1)

geef het aantal	regels in een file	<i>wc</i> (1)
print file op de	QUME printer	<i>qupr</i> (2)
print met grote,	samengestelde letters	<i>banner</i> (5)
	samenvoegen van gesorteerde files	<i>sort</i> (5)
	samenvoegen van ongesorteerde files	<i>cat</i> (1)
verstuur/ontvang	SARA files	<i>sara</i> (5)
verander lees- of	schrijfpermissie	<i>chmod</i> (4)
	sorteren van files	<i>sort</i> (5)
signaleer	spelfouten in engelse tekst	<i>spell</i> (1)
	splits file	<i>split</i> (5)
geef	staart van file	<i>tail</i> (5)
zoek bepaalde	string in een file	<i>grep</i> (1)
	systeemingang	<i>login</i> (5)
	toegang tot het systeem	<i>login</i> (5)
	tolereer inkomende write-berichten	<i>msg</i> (5)
	vercijfer of ontcijfer file	<i>crypt</i> (5)
	verbiedt inkomende write-berichten	<i>msg</i> (5)
	vergelijk files	<i>cmp</i> (1)
ga na of files	verschillen	<i>cmp</i> (1)
geef	verschillen tussen twee files	<i>diff</i> (1)
	verwijder file	<i>rm</i> (4)
	verwijder directory	<i>rmdir</i> (4)
	verzend gebruikers-post	<i>mail</i> (5)
	wie is er op het systeem	<i>who</i> (5)
zoek bepaald	woord in een file	<i>grep</i> (1)
geef het aantal	woorden in een file	<i>wc</i> (1)
	zoek in een file bepaalde regels	<i>grep</i> (1)
	zoek in een file bepaalde string	<i>grep</i> (1)
	zoek in een file bepaald woord	<i>grep</i> (1)

[illegible]

[illegible]

"this page is left blank intentionally"

1. Inleiding
 - 1.1. Opzet van de handleiding
 - 1.2. Overzicht van het systeem.
 - 1.3. Corrigeren tijdens het intypen.
 - 1.4. Beveiliging van files.
2. Het intypen van een tekst (Een voorbeeld).
 - 2.1. Het starten van het edit-programma.
 - 2.2. Het a-commando.
 - 2.3. Het print commando, regelnummers.
 - 2.4. Het w commando (W)
3. Het corrigeren van een tekst. (Een voorbeeld)
 - 3.1. Inleiding
 - 3.2. Het e-commando. (E)
 - 3.3. Het f-commando.
 - 3.4. Nogmaals: het a-commando.
 - 3.5. Het s-commando
 - 3.6. Het d-commando.
4. De fijne kneepjes.
 - 4.1. Inleiding.
 - 4.2. Het l-commando.
 - 4.3. Het m-commando.
 - 4.4. Het g-commando.
 - 4.5. Het r-commando.
 - 4.6. Details van het s-commando.
 - 4.6.1. Wijzigingen in meerdere regels tegelijk.
 - 4.6.2. Het &-teken in het s-commando.
 - 4.6.3. De symbolen en \$ in het s-commando.
 - 4.6.4. Het *-symbool in het s-commando.
 - 4.6.5. Het \-symbool in het s-commando.
 - 4.7. Het b-commando. (B)
 - 4.8. Het q-commando. (Q)

1. Inleiding

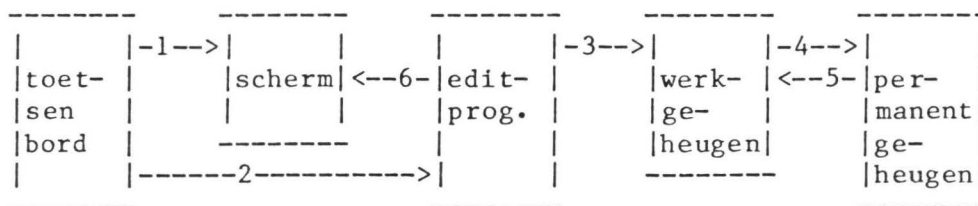
1.1. Opzet van de handleiding

Deze handleiding vertelt hoe vanaf de terminal teksten ingetypt en gecorrigeerd kunnen worden. De basis-kennis wordt besproken in het tweede deel van de inleiding (1.2) en de hoofdstukjes 2 en 3. In hoofdstuk 2 wordt aan de hand van een voorbeeld gedemonstreerd, hoe teksten ingetypt, en in de computer bewaard kunnen worden voor later gebruik. In hoofdstuk 3 wordt, ook weer met voorbeelden, het minimaal noodzakelijke over het corrigeren van teksten verteld. Tenslotte staan in hoofdstuk 4 de 'fijne kneepjes' van een aantal correctiemogelijkheden. Het is niet strikt nodig om precies te weten wat daar behandeld wordt, hoofdstukken 2 en 3 bevatten in principe voldoende informatie om te kunnen werken. Maar de trucjes uit hoofdstuk 4 kunnen wel veel werk besparen, dus kijk er eens naar, nadat hoofdstuk 2 en 3 echt duidelijk zijn.

Tenslotte: Het is niet de bedoeling deze handleiding alleen maar te lezen; ga er eens mee achter een terminal zitten en probeer de voorbeelden uit.

1.2. Overzicht van het systeem.

In dit paragraafje wordt een overzicht van het systeem gegeven, vooral de samenhang tussen toetsenbord, beeldscherm, edit-programma (kortweg: de 'editor') werkgeheugen en permanentgeheugen komen aan de orde. Misschien is het verhaal een beetje te abstract om meteen helemaal duidelijk te zijn. Dat is niet zo erg, in hoofdstuk 2 en 3 komen we er nog meermalen op terug.



```

*                               * *                               *
*   terminal                   * *   computer                   *
*****                         *****

```

De op het toetsenbord van de terminal ingetypte tekst komt in de eerste plaats op het scherm van de terminal terecht, zodat je kunt zien wat je getypt hebt (pijl 1). Ook de computer kan boodschappen naar het scherm van de terminal sturen (pijl 6).

Verder heeft het scherm van de terminal geen betekenis, het is alleen een soort mededelingenbord waarop te zien is, wat er vanaf de terminal is ingetypt en welke boodschappen de computer er naar toe heeft gestuurd.

Veel belangrijker is, dat ingetypte teksten door de terminal ook naar de computer gestuurd worden (pijl 2). In ons geval 'draait' in de computer het zg. edit-programma dat deze teksten aanpakt. Er is een verschil tussen de manier waarop teksten naar het scherm en naar het edit-programma gestuurd worden: Aanslagen van teksten verschijnen meteen op het scherm, het edit-programma krijgt alleen hele regels tekst te zien. Dus, zolang een ingetypte regel niet beëindigd wordt door een 'carriage return', staat hij wel is waar wel op het scherm, maar krijgt het edit-programma er geen letter van te zien.

Nogmaals, het edit-programma ontvangt dus volledige regels van de terminal. En er is dan sprake van twee verschillende soorten regels:

1. Commando-regels; deze bevatten een opdracht aan het edit-programma, b.v. om een bepaalde correctie uit te voeren.
2. Tekstregels; dit zijn regels van de eigenlijke in te typen tekst, die dus geen opdrachten aan het edit-programma bevatten. Deze tekstregels worden door het edit-programma aanpakkt, en doorgestuurd naar het zg. werkgeheugen (pijl 3) waar ze tijdelijk bewaard worden, zodat er dan met hulp van edit-commando's nog iets mee gedaan kan worden (bv. corrigeren).

Als het edit-programma gestopt zou worden, bv. omdat er een ander programma moet draaien of omdat de computer uitgezet wordt of een storing heeft, raakt de inhoud van het werkgeheugen (dus: de ingetypte tekst) verloren. Daarom bestaat er een edit-commando om het werkgeheugen te copieren naar het permanente geheugen (pijl 4). De inhoud van het permanente geheugen blijft bewaard,

ook als het programma of de computer gestopt wordt, en meestal ook bij storingen.

De inhoud van het permanente geheugen is voor het edit-programma ontoegankelijk, het is alleen maar een bewaarplaats. Het edit-programma heeft daarom de mogelijkheid om onder invloed van een aan de terminal ingetypte edit-commando teksten terug te halen van het permanente geheugen naar het werkgeheugen (pijl 5). Daarna kan er met deze teksten doorgewerkt worden.

1.3. Corrigeren tijdens het intypen.

Omdat het edit-programma alleen complete regels te zien krijgt, kunnen we tijdens het intypen van een regel al wat corrigeren!

Stel: we willen typen

'Boer er ligt een kip in het water',

en we hebben getypt

'Boer er ligt een kop in',

waarna we de fout ontdekken.

Op het scherm staat dan:

Boer er ligt een kop in

waarbij het streepje (de cursor) aan het eind van de regel door de terminmal neergezet is. Dit streepje geeft aan, op welke plek in de regel we aan het typen zijn. (hier dus aan het eind). We kunnen nu met de 'backspace'-toets dit streepje terug zetten tot onder de *o*, op het scherm staat dan:

Boer er ligt een kop in

Typen we nu 'ip in het water', dan staat er: *Boer er ligt een kip in het water*. Drukken we nu op de return toets, dan wordt de correcte regel overgestuurd. Het edit-programma heeft geen weet van de tussentijdse fout. Bevat de regel teveel fouten, dan kunnen we 'CNTL x' intypen, waarmee de regel tot dan toe wordt weggegooid, bv:

bv. Beex e! picht`CNTL x`Boer er ligt een kip in het water. Alleen de tekst na de 'CNTL x' wordt naar het edit-programma doorgestuurd. Er staat dus:

Boer er ligt een kip in het water.

1.4. Beveiliging van files.

Er zijn enkele speciale 'trucs' in het programma geïnstalleerd, waardoor u door buitengewone omstandigheden, down gaan en zo, minder kans heeft uw edit-file verloren te raken.

1. Indien de verbinding met de computer verbroken raakt, u tikt per ongeluk 3 breaks in binnen 3 seconden, dan worden alle programma's vanaf uw terminal met de computer, afgebroken. Het 'editor' programma schrijft uw file dan weg op de naam ed.hup dus kijk daar altijd even naar. Ook als u meer dan 30 minuten niets hebt ingetypt wordt de verbinding verbroken. Terminal uitzetten kan ook, enz.
2. Uw edit-file wordt eens in het half uur automatisch weggeschreven op een andere 'disk', ook het w-commando veroorzaakt dit. Dit is gemaakt opdat als de computer voor lange tijd echt kapot is, dat de files op een andere computer in te lezen zijn. Dus een beveiling die alleen toegepast wordt door de systeembeheerders, maar wel van ieders file.

Nb. de bescherming van uw tijdelijke edit-file is zodanig dat alleen u hem kan lezen en schrijven. NL de bescherming van uw weggeschreven file kunt u veranderen met het 'chmod' l) commando oid.

1) dit is geen 'editor' commando

2. Het intypen van een tekst (Een voorbeeld).

2.1. Het starten van het edit-programma.

2.2. Het a-commando.

Als het edit-programma gestart is, stuurt het naar het scherm van de terminal de tekst `c:` het programma vertelt hiermee, dat het klaar staat om een opdracht in ontvangst te nemen en uit te voeren. Wat we willen, is een tekst intypen, dus de opdracht die we willen geven luidt: 'Pak tot nader order de door mij ingetypte tekst aan, en berg die in het werkgeheugen'. We vertellen dit het edit-programma, door het commando `'a'` (van append) in te typen, gevolgd door een aanslag van de return toets, (Daardoor komt dit commando daadwerkelijk bij het edit-programma terecht). Het programma antwoordt dan met `l:` (van line), waarmee het om de eerste regel tekst vraagt.

Het beeldscherm ziet er nu als volgt uit: `c: a`
`l:`

Achter de `l:` kunnen we nu de eerste regel tekst intypen (afgesloten door return); het programma vraagt dan weer met een `l:` om de volgende regel, net zo lang totdat we er genoeg van hebben. We moeten op dat moment het programma vertellen, dat er geen regels tekst meer komen, d.i. dat de append-opdracht niet meer geldt. Dit gebeurt door een enkele `'.'` te typen in de plaats van een regel. De editor antwoordt dan niet meer met een `l:` maar met een `c:` waarmee het om een nieuw commando vraagt.

Na dit alles zou het scherm er als volgt uit kunnen zien:

```
c: a
l: De invoering van automatisering,
l: die tegenwoordig op grote schaal plaats vindt,
l: kan vergeleken worden met de introductie
l: van de stoommachine in vroeger tijden.
l: net als toen, kunnen ook nu weer
l: de gevaren niet afgewend worden.
l: door machines stuk te slaan.
l: .
door machines stuk te slaan.
c: _
```

We zien, dat de laatste regel twee keer op het scherm staat, een keer voor, en een keer na de punt. Dit komt omdat elk commando zijn resultaat in een of andere vorm afdruckt. Het append commando laat als resultaat de laatst toegevoegde regel zien. (alle regels zou ook gekund hebben, maar dat wordt bij lange teksten gauw erg vervelend.) Waarom juist de laatste regel afgedrukt wordt en niet bv. de eerste of eentje ergens in het midden, zal nog blijken.

2.3. Het print commando, regelnummers.

De tekst van het voorbeeldje is zo kort, dat hij in zijn geheel op het scherm past, we kunnen dus alles wat we getypt hebben nog zien. Teksten, langer dan 24 regels passen niet meer op het scherm. Om ze te kunnen inspecteren, hebben we het 'p' (print)-commando.

De tekstregels in het werkgeheugen worden door het edit-programma a.h.w. genummerd; deze nummering kunnen we gebruiken om selectief regels af te drukken. Het commando 'lp' drukt de eerste regel af, '6p' de zesde en '2,4p' de tweede tot en met de 4de. Op het scherm zou, als deze commando's achtereenvolgend gegeven worden, het volgende te zien zijn:

```
c: lp
De invoering van automatisering,
c: 6p
de gevaren niet afgewend worden.
c: 2,4p
die tegenwoordig op grote schaal plaatsvindt,
kan vergeleken worden met de introductie
van de stoommachine in vroeger tijden.
c:
```

Het probleem met deze regelnummering is, dat we ze vaak niet kennen. Wat is bv. het nummer van de laatste regel van een lange tekst? En hoe kunnen we die regel op het scherm krijgen, zonder eerst alle regels te tellen? Het antwoord is, dat overal waar we het nummer van de laatste regel zouden willen gebruiken, we ook het '\$'-symbool kunnen typen. De editor weet dan, dat de laatste regel bedoeld is, dus: '\$p' drukt de laatste regel af en 'l,\$p' vanaf de eerste tot de laatste regel, d.i. de hele tekst.

Het edit-programma onthoudt het nummer van de laatst afgedrukte regel, we noemen hem: huidige regel. Net zoals de '\$' de laatste regel aanduidt, duidt het '.'-symbool de huidige regel aan, het commando '.p' drukt de huidige regel af. Willen we het echte nummer van de huidige regel te weten komen, dan kunnen we het '='-commando gebruiken. '=' geeft het nummer van de huidige regel, '\$=' werkt ook: het geeft het nummer van de laatste regel.

Geven we het commando `4p`, dan wordt de 4e regel afgedrukt, dit is dan meteen de huidige regel. Het commando `.p` drukt deze regel dan opnieuw af. Geven we het commando `.=`, dan antwoordt de editor met 4 het regelnummer. In het geval van het voorbeeld zouden we op het scherm te zien krijgen :

```
c: 4p
van de stoommachine in vroeger tijden.
c: .p
van de stoommachine in vroeger tijden.
c: .=
4
c:
```

Tenslotte kan de editor een beetje met regelnummer rekenen. Het commando `.-2,.+lp` drukt de regels af vanaf 2 regels voor de huidige, tot 1 regel na de huidige. In het voorbeeld zouden dit de regels 2 t/m 5 zijn (de huidige regel was 4), waarna de huidige regel, regel 5 is geworden (de laatst afgedrukte).

2.4. Het w commando

Het w commando zorgt ervoor dat tekst van het werkgeheugen gecopieerd wordt (wordt weggeschreven) naar het permanente geheugen. Zo'n gecopieerde tekst heet een `file`. Omdat het permanente geheugen vele files kan herbergen, en we files ieder afzonderlijk terug willen kunnen halen naar het werkgeheugen om er mee door te werken (zie hoofdstuk 3), moeten we files d.m.v. het `w` commando namen geven.

In het geval van ons voorbeeld zouden we de tekst bv. voorbl willen noemen (we willen een file hebben, met file-naam voorbl). We geven dan het commando: `w voorbl`. Op het scherm:

```
c: w voorbl
258
c: _
```

Het op het scherm afgedrukte `resultaat` van het w commando is een getal, hier 258. Het is het aantal lettertekens, waaruit de weggeschreven tekst en dus de in het permanente geheugen gecreëerde file, bestaat (NB: ook tabs, returns, spaties,, etc. noemen we hier lettertekens).

Ook het w commando kent regelnummers, hiermee kunnen we teksten splitsen. Geven we bv. in het geval van het voorbeeld de commando's `1,4w voorb2` en `3,\$w voorb3`, dan maken we daarmee de files voorb2 en voorb3, die de eerste 4 regels, resp. de 3e t/m de laatste regel van de tekst bevatten.

Als de ingetypte tekst bewaard moet worden, moet - voordat de terminal verlaten wordt - altijd het w commando gegeven worden. Deze tekst verdwijnt nl. uit het werkgeheugen als de computer gestopt wordt, het edit-programma gestopt wordt, of als iemand anders vanaf dezelfde terminal gaat editen. Een tekst in het permanente geheugen overleeft deze gebeurtenissen wel.

Met het w commando moeten we oppassen. Als we bv. het commando 'w voorbl' geven, terwijl er al een file met naam voorbl bestaat, dan vraagt het editor programma of je dat echt bedoeld, je moet dan antwoorden met 'y' anders wordt de file niet weggeschreven.

3. Het corrigeren van een tekst. (Een voorbeeld)

3.1. Inleiding

Een tekst kan alleen gecorrigeerd worden als hij zich in het werkgeheugen van de computer bevindt. Als we hem net hebben ingetypt, zit hij daar nog en kunnen we meteen corrigeren (of voortgaan met intypen). Anders (bv. als de terminal tussentijds door anderen gebruikt is), moeten we hem eerst van het permanente geheugen terughalen naar het werkgeheugen. Hiervoor dient 'e' (edit)-commando.

3.2. Het e-commando.

Het edit-commando dient om files uit het permanente geheugen terug te halen naar het werkgeheugen. Welke file we willen terughalen wordt aangegeven door de naam die de file (bv. d.m.v. het w-commando) eerder gekregen heeft. Zo halen de commando's 'e voorbl', 'e voorb2', resp. 'e voorb3', de files voorbl, voorb2 resp. voorb3 (zie par.2.4) naar het werkgeheugen. (Niet allemaal tegelijk, de vorige file wordt vervangen door de volgende!)

Op het scherm ziet dit er als volgt uit:

c: e voorbl

258

De invoering van automatisering

c: _

Het e-commando geeft dus als 'resultaat' op het scherm:

1. Het aantal lettertekens waaruit de file bestaat. Dit is natuurlijk hetzelfde aantal als het w-commando gaf toen de file werd weggeschreven.
 2. De eerste regel van de file.
- De afgedrukte eerste regel is meteen de huidige regel! Zouden we nu dus het command 'p' geven, dan kwam de eerste regel opnieuw op het scherm!

Met het edit-commando moeten we voorzichtig zijn! Een commando 'e file' leidt er nl. toe dat de inhoud van het werkgeheugen vervangen wordt door die van de file. Als we de oude tekst niet kwijt willen raken, moet hij dus altijd eerst worden weggeschreven, voordat we het 'e-commando' gebruiken.

Mocht dit toch gebeuren en we hebben de oude file gewijzigd, dan antwoord het 'editor' programma met *No!*

?

c:

3.3. Het f-commando.

Met het f-commando kunnen we het edit-programma vragen hoe de file 'die we het laatst in de editor gehaald hebben' (dwz. met het e-commando naar het werkgeheugen gehaald hebben) ook alweer heet. Op het scherm:

c: e voorbl

258

c: f

voorbl

c: _

Dit is alleen maar handig, als we de naam van de file die we editen, even vergeten zijn!

3.4. Nogmaals: het a-commando.

We hebben het a-commando al in hoofdstuk 2 gezien, er kan echter meer mee. We kunnen er ook in een al eerder ingetypte tekst regels mee invoegen of er aan het eind regels aan toevoegen. De truc is, het append commando vooraf te laten gaan door het regelnummer van de tekstregel, waarachter we willen invoegen (toevoegen). Willen we bv. in de tekst van voorbl na regel 4 iets toevoegen, dan geven we het commando: 4a. Op het scherm kan dit er bv. als volgt uitzien:

c: e voorbl

c: 4a

l: Net als toen, is er ook nu

l: angst voor ontslagen en weerstand

l: tegen verandering van de inhoud van het werk

l: .

weerstand tegen verandering van de inhoud van het werk

c: _

In plaats van '4a' hadden we ook achtereenvolgend de commando's '4p' en '.a' kunnen geven. '4p' maakt van de vierde regel nl. de huidige regel, en '.a' voegt toe achter de huidige regel.

Willen we regels achter de tekst toevoegen, dan zeggen we
`\$a` (voeg toe achter de laatste regel), bv.:

```
c: $a
l: Dit zou bovendien gevaarlijk zijn
l: zeker als de stekker van de terminal
l: nog in het stopcontact zit.
l: .
nog in het stopcontact zit.
```

Regels voor de eerste regel toevoegen gaat met een trucje;
we zeggen `0a`, d.i.: voeg toe achter de 0-de regel. bv.

```
c: 0a
l: Een voorbeeld.
l: .
```

Na al deze commando's zou onze tekst er als volgt uitzien:
(we kunnen dit bekijken met het commando: l,\$p)

```
Een voorbeeld.
De invoering van automatisering
die tegenwoordig op grote schaal plaats vindt,
kan vergeleken worden met de introductie
van de stoommachine in vroegertijden
Net als toen, is er ook nu
angst voor ontslagen en weerstand
tegen verandering van de inhoud van het werk
Net als toen, kunnen er ook nu weer
de gevaren niet afgewend worden
door machines stuk te slaan
Dit zou bovendien gevaarlijk zijn,
zeker als de stekker van de terminal
nog in het stopcontact zit.
c:
```

3.5. Het s-commando

Stel eens dat het volgende stukje tekst per vergissing is
ingetypt:

```
aal
neet
muil
wi m
```

We willen dit vervangen door:

```
aap
```

```
noot
mies
wim
```

Dit kan met het `s` (substitute-, d.i. vervang door-, commando), waarmee we stukken tekst binnen een regel kunnen vervangen door andere. Laten we beginnen met het woord `aal`; hieronder een voorbeeld van hoe die vervanging to stand kan komen.

```
c: lp
aal
c: s/l/p/
aap
c: _
```

We gebruiken eerst het commando `'lp'` om de regel waarin we willen wijzigen (hier de eerste), tot de huidige regel te maken. Daarna komt het `s`-commando, dat hier betekent: 'vervang in de huidige regel de letter `l` door de letter `p`'. Hierna drukt het `s`-commando de gewijzigde regel af, d.i. `aap`.

Dit had ook korter gekund, nl.:

```
c: ls/l/p/
aap
c:
```

Het `s`-commando begint hier met het regelnummer van de regel die gewijzigd moet worden, dit spaart een `'p'`-commando uit. Maar, met regelnummers kunnen gemakkelijk vergissingen ontstaan. Eerst afdrukken om te verifiëren of we de goede regel te pakken hebben, en daarna `s` gebruiken, geeft minder fouten.

We vervangen nu `neet` door `'noot'`, dit zou als volgt kunnen:

```
c: 2p
niet
c: .s/e/o/
noet
c: .s/e/o/
noot
c:
```

Dus: het `s`-commando vervangt steeds de eerste in aanmerking komende letter in de regel door een andere; de eerste `e` uit `neet` wordt andere `e` te vervangen. Dit kan sneller:

```
c: 2p
neet
c: .sg/e/o/
noot
```


Het ``sg'`-commando werkt net als het `s`-commando, alleen voert het de wijziging op alle mogelijke plaatsen in de regel uit! (g staat voor ``global'`) Nu de wijziging van *muil* in *`mies'*.

```
c: 3p
muil
c: s/uil/ies/
mies
c:
```

Het is dus niet nodig letter voor letter te wijzigen (hier *u* door *i*, *i* door *e*, *l* door *s*). Het `s`-commando kan in een keer stukken van regels vervangen door andere. Ten slotte willen we *wi m* vervangen door *wim*, d.i. de spatie weghalen.

```
c: 4p
wi m
c: .s/ //
wim
c:
```

Dit ziet er misschien wat raar uit, wat er gebeurt is het volgende: De tekst tussen de eerste en tweede ``/'` wordt zoals altijd vervangen door de tekst tussen de tweede en derde ``/'` uit het commando. Tussen de eerste en tweede ``/'` staat hier een spatie, tussen de tweede en derde ``/'` staat geen tekst. De spatie wordt dus vervangen door ``geen tekst'`, d.i. hij verdwijnt.

3.6. Het `d`-commando.

Met het ``d'` (delete)-commando kunnen we regels tekst weggomen. ``d'` gooit de huidige regel weg, `ld` de eerste, `l0d` de 10e, `3,5d` de 3e t/m de 5e en `l,$d` alles.

In het volgende voorbeeld gaan we ervan uit, dat we nog steeds met de *aap*, *noot*, *etc.* -file werken.

```
c: 2,3d
wim
c: .d
?
c: l,$p
aap
```

Eerst worden de regels 2 en 3 verwijderd, het commando antwoordt met de eerstvolgende niet-verwijderde regel, dit is hier de 4e regel, *`wim'*. Dit wordt dan meteen de huidige regel, zodat ``d'` nu deze regel weggooit. Nu wordt geantwoord met een `?` omdat *wim* de laatste regel van de file was. Er is dus geen eerst-

volgende nog aanwezige regel om te vertonen en tot de huidige regel te maken. De regel *wim* is echter wel weggegooid, zodat *l,\$p* (druk alle regels af) alleen *aap* oplevert.

Het delete-commando is nogal gevaarlijk, iets dat per vergissing verwijderd is, zal opnieuw moeten worden ingetypt. Een veiliger vorm is het dv-commando (delete en veto). Dit commando laat alle te verwijderen regels een voor een op het scherm zien. Antwoorden we met 'n' dan wordt deze regel niet verwijderd, als we 'return' aanslaan, wel. Andere antwoorden worden door de editor op zijn beurt beantwoord met '?', waarna we opnieuw de keus hebben tussen 'n' en return.

In het 'aap, noot,...'-voorbeeld (aangenomen dat er nog niets is weggegooid, kan dit er als volgt uitzien:

```
c: 1,3dv
aap
```

```
noot
n
mies
```

```
wim
c: 1,$p
noot
wim
c:
```

Eerst wordt regel 1 *aap* vertoont. We antwoorden met een return (in het voorbeeld te zien als blanke scherm-regel). *noot* willen we niet weggooien, we antwoorden met 'n', maar *mies* mag weer wel weg. Het commando antwoordt weer met de eerstvolgende niet verwijderde regel, nl. *wim*. 'l,\$p' vertoont nu de resterende regels, nl. *noot* en *wim*.

Opmerking:

Stel, we hebben een bepaalde tekst ingetypt EN WEGGESCHREVEN, (w-commando) en willen nu een andere tekst gaan intypen. We moeten dan eerst het commando *l,\$d* geven om de oude tekst uit het werkgeheugen te verwijderen, anders zou de nieuwe tekst aan de oude vastgeplakt worden. PAS OP, DAT VOOR l,\$d DE OUDE TEKST IS WEGGESCHREVEN, anders gaat hij verloren!

4. De fijne kneepjes.

4.1. Inleiding.

Met de tot dusverre behandelde commando's is het intypen en corrigeren van tekst volledig mogelijk. De in dit hoofdstuk besproken commando's zijn dus strikt gesproken niet nodig. Ze kunnen echter wel erg veel werk besparen.

4.2. Het l-commando.

Stel, ergens midden in een tekst zit een typefout, een regel eindigt met *raar.* terwijl er *paar.* had moeten staan. Voordat we met het s-commando kunnen corrigeren, moeten we het regelnummer van de foutieve regel kennen. Dit is met de 'p' en '='-commando's (zie 2.3) wel te achterhalen, maar dat is bij lange teksten heel wat werk.

Het l-commando (locate-commando) kan uitkomst brengen, in ons voorbeeld bv.:

c: ll/raar./

112

Twee schoenen noemt men ook wel een raar.

c: _

'll/raar/' betekent het volgende: Zoek, beginnend bij de eerste regel, de eerstvolgende regel op, die de tekst *raar.* bevat. Druk het nummer van die regel, en de regel zelf af.

Weten we zeker, dat de gezochte regel zich achter de huidige bevindt, dan typen we natuurlijk '.l/.../' i.p.v. ll/.../. I.h.a. mag voor de l een willekeurig regelnummer staan.

Heeft het l-commando tot aan het eind van de tekst gezocht zonder een in aanmerking komende regel te vinden, dan wordt geantwoord met '?'.

Vindt het l-command een regel die de gewenste tekst bevat, maar is dat niet de door ons bedoelde (een tekst zoals *raar.* kan in meerdere regels voorkomen), dan kunnen we vanaf de gevonden regel doorzoeken door kortweg in te typen:

l//

De editor zoekt dan verder, met de de laatste keer tussen '/''s opgegeven tekst, vanaf de laatsgevonden regel!

We kunnen echter i.p.v. van voren naar achteren ook van achteren naar voren zoeken. We hoeven dan alleen maar de '/' te vervangen door een '?'. Dus:

c: .l?raar.?

18

raar.

c:

4.3. Het m-commando.

Het m(move)-commando verwisselt stukken tekst. Stel de tekst is weer

aap

noot

mies

wim.

Het commando: 2m3 plaats de 2e regel achter de 3e, de tekst wordt: aap

mies

noot

wim.

Het commando 1,2m\$ plaatst de eerste 2 regels achter de laatste, de tekst wordt:

noot

wim

aap

mies

Het m-commando maakt van de huidige regel de laatst-verplaatste, en drukt deze af. Het 2m3-commando uit bovenstaand voorbeeld zou dus op het scherm het volgende te zien geven:

c: 2m3

noot

c:

Het regelnummer van de huidige regel is hier 3, omdat noot de derde regel geworden is.

4.4. Het g-commando.

Het g(glue)-commando kan gebruikt worden om twee opeenvolgende tekstregels te vervangen door een enkele. Deze bestaat uit de twee oorspronkelijke achter elkaar, gescheiden door een spatie. '10g' werkt op de 10e en 11e regel, '.g' op de huidige en

de daarop volgende. Het g-commando drukt de aldus samengestelde regel op het scherm af. Dit wordt tevens meteen de huidige regel. Stel, we hebben de volgende tekst: *De eerste regel en de tweede en de derde*. Het commando 2g verandert deze tekst in:

De eerste regel

en de tweede en de derde.

Op het scherm zouden we te zien krijgen:

c: 2g

en de tweede en de derde.

c:

4.5. Het r-commando.

Het r-commando (read commando) kan net als het e commando, gebruikt worden om tekst vanuit het permanente geheugen in het werkgeheugen te halen. Het verschil is, dat bij het e-commando de tekst in het werkgeheugen vervangen wordt, bij het r-commando wordt de ingelezen tekst toegevoegd aan de al aanwezige.

Het r-commando moet weer voorafgegaan worden door een regelnummer. De ingelezen tekst wordt ingevoegd tussen de regel met dit nummer en de daarop volgende.

Het commando `10r filenaam' volgt de tekst van file `filenaam' in tussen regel 10 en 11 van de tekst uit het werkgeheugen. Het commando `\$r filenaam' voegt de tekst toe, achter de laatste regel.

Het r-command antwoordt met een getal (wederom het aantal ingelezen symbolen) en met de eerste ingelezen regel. Dit wordt tevens de huidige regel.

Voorbeeld:

c: a

l: Een regel

l: En nog een

l: .

En nog een

c: w file1

21

c: 1,\$d

?

c: a

l: Een tussenvoeging

l: .

Een tussenvoeging

c: w file2

18

c: e file1

21

Een regel

```
c: lr file 2
18
Een tussenvoeging
1,$p
Een regel
Een tussenvoeging
En nog een
c:
```

Hier is het volgende gebeurd. Er is een tekst aangemaakt, die is weggeschreven naar file1, daarna is het werkgeheugen leeg gemaakt (1,\$d) waarna een nieuwe tekst is aangemaakt op file2. Dan wordt file1 weer in de editor gehaald en file 2 achter de eerste regel hiervan ingelezen. Tenslotte wordt het resultaat op het scherm afgedrukt.

Het r-commando wordt gebruikt, om zich op meerdere files bevindende teksten samen te voegen tot een enkele.

4.6. Details van het s-commando.

4.6.1. Wijzigingen in meerdere regels tegelijk.

Het s-commando kan op meerdere regels tegelijk werken. Zo vervangt '1,10s/A/B/' de eerste letter A in de eerste 10 regels door een B.

Systematische typefouten kunnen we proberen in een keer in de hele tekst te corrigeren. Stel, in een tekst is overal *licht* geschreven i.p.v. 'ligt'. 1,\$sg/licht/ligt/, vervangt in elke regel op elke plek *licht* door *ligt*. We moeten hier erg mee oppassen omdat ook delen van woorden die toevallig de opeenvolgende letters *licht* bevatten, veranderd worden. Zo wordt: *hij licht* vervangen door *hij ligt*, maar ook *lichtschip* door *ligtschip*!! Deze laatste fout wordt voorkomen door het meegeven van spaties, d.i.: 1,\$s/ licht / ligt /. Woorden die de letters l-i-c-h-t bevatten, worden nu niet aangetast. Maar,... het woord *licht* aan het begin

of het eind van een regel wordt nu niet veranderd.

4.6.2. Het &-teken in het s-commando.

Stel, we hebben een tekstregel die luidt: *Deze volg moet veranderd worden*. In plaats van *volg* had er natuurlijk moeten staan: 'volgorde'. Het commando 's/volg/volgorde/' voert de correctie uit, maar ... we moeten twee keer achter elkaar 'volgt' typen. Kortser is: 's/volg/&orde/' Het teken '&' tussen de tweede en derde '/' zorgt ervoor, dat op die plek de tekst die zich tussen de eerste en tweede '/' bevindt, gebruikt wordt (Hier dus 'volgt'). Dit scheelt typewerk.

4.6.3. De symbolen ^ en \$ in het s-commando.

Het ^-teken geeft het begin van de regel, het \$-teken het eind van de regel aan. '.s/^/huis /' voegt 'huis ' toe aan het begin van de huidige regel. '.s/\$/ huis/' voegt 'huis' toe aan het eind van de regel. Merk op, dat we in het eerste geval een spatie op 'huis' hebben laten volgen en in het tweede geval eraan laten voorafgaan. Zonder dat, zou het woord 'huis' aan het eerste resp. laatste woord van de regel zijn vastgeplakt!

4.6.4. Het *-symbool in het s-commando.

Het '*' symbool staat voor een willekeurige rij opeenvolgende tekens. Moeten we bv. *txyz?)(n* vervangen door 'tuinman', dan kan dit met: *.s/t*n/tuinman/*. Ook dit scheelt weer typewerk. Maar: de '*' slaat (in dit voorbeeld) op de eerste 't' en de laatste 'n' in de regel, dus: *drinken we thee met de txyz)(n uit nappen? nee!* zou met dit commando vervangen worden door: *drinken we tuinmanee.*

Oppassen dus!

4.6.5. Het \-symbool in het s-commando.

We hebben gezien, dat `'/'`, `'$'`, `'*'`, `'^'`, en `'&'` speciale betekenis hebben. Als deze tekens 'gewoon' in de tekst voorkomen, en we willen ze wijzigen, ontstaan problemen. Stel bv `$100.000` moet vervangen worden door `'fl.100.000'`. Het commando `s/$/fl./` produceert echter `$100.000fl.`, immers, de `'$'` betekent in het `'s'` commando 'het einde van de regel'. `s/\$/fl./` heeft wel het gewenste resultaat. De `'\''` voor de `'$'` betekent nl: 'deze `'$'` heeft geen speciale betekenis, maar is een gewoon tekst-symbool!.

Ook de andere speciale symbolen kunnen, door er een `'\''` voor te typen, van hun speciale betekenis ontdaan worden en als gewone tekst-tekens behandeld worden.

Het zelfde geldt voor het nieuwe speciale symbool `'\'\''` zelf. Moet dit gewijzigd worden, dan moet het door zichzelf voorafgegaan worden. bv: `'.s/\'\'\'/\'\'\'/'` vervangt `'\'\''` door `'/'`, bv in `'en\of'` door `'en/of'`.

4.7. Het b-commando en het v-commando. (B- en V-)

Met dit commando wordt het prompt-symbool niet afgedrukt. Dus zowel `c:` als `l:` verschijnen niet op het scherm. De oude situatie wordt weer hersteld met het v-commando.

Het b-commando is handig bij het gebruik van tabs in een file.

Nb. het b-commando geeft de melding: *No prompt symbol until the command 'v'* dit wordt niet gedaan met het B-commando. Het V-commando is identiek aan het v-commando.

4.8. Het q- en het Q-commando.

Met dit commando verlaat u de 'editor' en de computer antwoordt met het standaard prompt symbool, meestal het `%` teken. Als uw file niet is weggeschreven, of is veranderd sinds de laatste `'w'` opdracht, dan weigert 'editor' deze opdracht te vervullen. U kunt toch geforceerd te 'editor' verlaten door het Q-commando. U kunt daardoor echter informatie verloren laten gaan.

SUMMARY v/d commando's

- 3.2 e filename
haal de file genaamd 'filename' in de editor.
- 3.3 f
vraag de naam van de file die in de editor is.
- 3.4 a
[nr]a
append vanaf regelnummer aangeduid met nr.
indien geen [nr] dan wordt regel '.' verondersteld.
ipv [nr] mag ook '.' '-' '+' '\$' enz.
- 3.5 s
s/string1/string2/
vervang 'string1' door 'string2'
- 3.6 d
[range]d
[range] kan zijn: [nr] of [nr1],[nr2]
verwijder regel, indien geen [range] gespecificeerd
dan wordt regel '.' verondersteld.
- 2.3 [range]p
print regel(s)
- 2.4 w filename
write commando, schrijf een editfile weg met de naam
'filename', er is bescherming tegen overschrijving van
al bestaande files.
- 4.2 l/string/
l?string?
zoek de regel waarin in 'string' voorkomt.
/string/ betekent: zoek vooruit vanwaar je nu bent.
?string? betekent: zoek achteruit vanwaar je nu bent.
- 4.3 [range]m[nr]
move, verplaats een antal regelnummers achter de regel
aangeduid met [nr].
- 4.4 [nr]g
glue commando geeft het volgende effect:
dit is een
stuk tekst

•g
dit is een stuk tekst
- 4.5 [nr]r filename
plaats achter regel [nr] de inhoud van de file aange-
duid met 'filename'.
- 4.6.3 en \$ symbool
in een string heeft het de betekenis van
begin van de regel en \$ het einde van de regel.
- 4.6.4 * symbool
verleng de string met alle karakters die verder op die-
zelfde regel voorkomen.
- 4.6.5 \ symbool
sommige karakters hebben een speciale betekenis zoals

spatie

\$

\ /

* &

om ze toch fatsoenlijk in de tekst op te nemen, moet je ze vooraf laten gaan door een \ (backslash).

4.7 b commando

print `c:` en `l:` niet af.

4.8 q commando

verlaat de `editor`.