Zelftest: Allerlei info en Commando's

1. Leg uit wat het verschil is tussen statisch en dynamisch linken? Enkel de uitleg volstaat.

Bij statisch linken worden alle nodige bibliotheken om een programma uit te voeren bij het compileren toegevoegd aan het programma. Bij dynamisch linken worden de bibliotheken opgezocht op het moment dat het programma wordt uitgevoerd. Bij statisch linken zal de bestandsgrootte van het programma groter zijn.

2. Hoe kan je een terminal venster leeg maken?

Met het commando `clear`

3. Hoe kan je met het commando man informatie opvragen over de informatie in het bestand /etc/passwd?

In [ ]:
```
%%bash
man 5 passwd
```

4. Met het commando man kan je informatie opvragen over externe Linux/Unix-opdrachten. Hoe kan je best info opvragen over interne opdrachten zoals echo, printf, set, cd, ...

Je kan de interne opdrachten opvragen met het commando `help` gevolgd door de opdracht waarover je informatie wil. Bijvoorbeeld `help echo`. Maar je kan ook man pages van de opdracht zelf opvragen.

5. Welk commando kun je gebruiken om:
   - gegevens te sorteren? `sort`
   - het verschil tussen twee bestanden te bekijken? `diff <bestand1> <bestand2>`
   - een bestand of directory te zoeken? `find`
   - duplicate regels uit een bestand te verwijderen? `sort | uniq -d`
   - enkel de 10 eerste lijnen uit een bestand op het scherm te tonen? `head -n 10`
   - enkel de 10 laatste lijnen uit een bestand op het scherm te tonen? `tail -n 10`
   - een string te zoeken in een tekstbestand? `grep <string> <bestand>`
   - het aantal lijnen, woorden en karakters van een bestand te tonen? `wc <bestand>`

6. De inhoud van de /dev-map bevat bestanden die je in twee groepen kan onderverdelen. Block special device files voor harde schijven en andere "mass storage devices" en character special device files voor de overige randapparatuur waar een device-node

voor werd voorzien (niet elk device heeft immers een device node). Wat is het essentiële verschil tussen een block- en character special device files?

Een block special device file is een bestand dat toegang geeft tot een blok van data, terwijl een character special device file toegang geeft tot een karakter van data.

7. Wat is de betekenis van de twee getallen die naast een device node vermeld staan?

Het eerste getal is de major device number en het tweede getal is de minor device number. De major device number geeft aan welke driver het apparaat bestuurt, terwijl de minor device number aangeeft welk apparaat het is.

8. Gebruik het commando find om een overzicht te krijgen van zowel de character special device files als de block special device files. Probeer het eerst door twee opdrachten te geven en daarna door de twee opdrachten te groeperen in één opdracht.

In [ ]:
```
%%bash
find /dev -type c
find /dev -type b
```

In [ ]:
```
%%bash
find /dev -type c -o -type b
```

9. Leg uit wat het verschil is tussen een hard link en een soft link. Wat zijn de voor- en nadelen van beide.

Een hardlink wijst altijd van een bestandsnaam naar data op een schijf. Een softlink wijst van een bestandsnaam naar een andere bestandsnaam, die op zijn beurt naar data op een schijf wijst.

- Voordelen hardlink: je kan een bestand verwijderen zonder dat de data verloren gaat.
- Nadelen hardlink: je kan geen hardlink maken naar een directory.
- Voordelen softlink: je kan een softlink maken naar een directory.
- Nadelen softlink: als het originele bestand wordt verwijderd, is de softlink niet meer geldig.

10. Maak in je home-directory een softlink aan naar /etc/passwd en een hardlink naar /etc/group (probeer dit met het commando cp als met het commando ln). Hoe kan je zien of het wel degelijk over een hardlink gaat en niet over een kopie?

In [ ]:
```
%%bash
cp -l /etc/group /home
cp -s /etc/passwd /home
```

```
ln /etc/group /home
ln -s /etc/passwd /home
```

Je kan zien of het over een hardlink gaat door het commando `ls -l` uit te voeren. Als het over een hardlink gaat, zal het aantal links naar het bestand verhoogd zijn.

11. Waarom is een pseudo random generator met een beperkt aantal bits geen goede random generator?

Een pseudo random generator met een beperkt aantal bits is geen goede random generator omdat de gegenereerde getallen zich herhalen na een bepaald aantal iteraties. Het is dus niet echt random.

12. Hoe kan je met het commando head en bijhorende optie 4 bytes uit /dev/random halen en deze bytes gewoon zonder te converteren naar het scherm schrijven?

In [ ]:
```
%%bash
head --bytes 4 /dev/random
```

13. Wat is pathname expansion? Wat zijn de verschillende metatekens die je bij pathname expansion kan gebruiken?

Pathname expansion is het proces waarbij een shell een patroon uitbreidt tot een lijst van bestandsnamen. De metatekens die je kan gebruiken zijn `*`, `?`, `[ ]`, `{ }`.

14. Hoe kan je met het commando “ls -l” een overzicht geven van alle bestandsnamen die bestaan uit minstens twee letters gevolgd door een cijfer gevolgd door een willekeurig aantal karakters?

In [ ]:
```
%%bash
ls -l [a-z][a,z][0-9]*
```

15. Wanneer je “echo *” ingeeft, welk proces zorgt dan dat die * wordt omgezet naar bestandsnamen?

Bash zelf. Het proces genaamd pathname expansion zorgt ervoor dat de * wordt omgezet naar bestandsnamen. (zie vraag 13)

16. Maak met touch een bestand aan met als naam passwd in je huidige werkdirectory. Zoek nu met het commando find naar alle bestands- en directorynamen, te beginnen bij /, die beginnen met het woord pass gevolgd door 0 of meerdere willekeurige tekens? Zorg dat find hier het *-teken omzet naar 0 of meerdere willekeurige tekens.

In [ ]:
```
%%bash
touch passwd
find . -type f -name "pass*"
```

Quotes zijn belangrijk want die pass* is voor find bedoeld en mag niet door Bash omgezet worden via pathname expansion wat zal gebeuren want je hebt met touch een bestand aangemaakt met als naam passwd. Dus bij u zal pass* vervangen worden door passwd en bij mij blijft het pass*

17. Genereer met brace expansion alle hexadecimale getallen van 00 tot FF.

In [ ]:
```
%%bash
echo {{0..9},{A..F}}{{0..9},{A..F}}
```

18. Wat doet het commando sync? Wat wordt er bedoeld met mounten en unmounten?

Het commando sync zorgt ervoor dat alle buffers naar de schijf worden geschreven (tijdelijke bestanden worden permanent gemaakt). Mounten is het koppelen van een bestandssysteem aan een directory in de bestandsstructuur. Unmounten is het ontkoppelen van een bestandssysteem van een directory in de bestandsstructuur.

19. Welke filedescriptoren worden er gebruikt voor standaard invoer, standaard uitvoer en het standaardfoutenkanaal? Wat is het essentiële verschil tussen de twee uitvoerkanalen?

Voor standaardinvoer wordt filedescriptor 0 gebruikt, voor standaarduitvoer wordt filedescriptor 1 gebruikt en voor het standaardfoutenkanaal wordt filedescriptor 2 gebruikt. Het verschil tussen standaarduitvoer en het standaardfoutenkanaal is dat standaarduitvoer bedoeld is voor normale uitvoer, terwijl het standaardfoutenkanaal bedoeld is voor foutmeldingen. ~ Vb: `ls 1> bestand` zal de uitvoer van ls naar bestand schrijven, terwijl `ls 2> bestand` de foutmeldingen van ls naar bestand schrijft.

20. Waarvoor dient /dev/null? Hoe kan je de gebufferde uitvoer van een willekeurige opdracht naar /dev/null schrijven? Hoe kan je de niet-gebufferde uitvoer van een willekeurige opdracht naar /dev/null schrijven? Hoe kan je ervoor zorgen dat ze nu allebeide naar /dev/null worden gestuurd?

De /dev/null is een speciaal bestand dat alle data die ernaar wordt geschreven, negeert.

- Gebufferde uitvoer naar /dev/null schrijven: `opdracht > /dev/null`
- Niet-gebufferde uitvoer naar /dev/null schrijven: `opdracht 2> /dev/null`
- Beide naar /dev/null schrijven: `opdracht > /dev/null 2>&1`

21. Wanneer is het handig om gebufferde uitvoer om te zetten naar niet gebufferde uitvoer? Wanneer is het nodig om niet-gebufferde uitvoer naar gebufferde uitvoer om te zetten?

- Niet gebufferd naar gebufferd: goed voor pipes, aangezien enkel gebufferde uitvoer door een pipe kan gaan, aangezien een pipe een queue gebruikt
- Gebufferd naar niet-gebufferd: goed voor foutmeldingen, aangezien foutmeldingen onmiddellijk moeten worden weergegeven

22. Wat is een pipe? Wat wordt doorgelaten en wat niet?

Een pipe is een manier om de uitvoer van een programma / commando door te geven aan een ander programma / commando. Alles wat naar standaarduitvoer wordt geschreven, wordt doorgelaten. Foutmeldingen worden niet doorgelaten.

23. Tel hoeveel fouten het commando “du /proc” oplevert en maak hierbij gebruik van een pipe?

In [ ]:
```
%%bash
du /proc 2>&1 | wc -l
```

24. Gegeven “strace shuf -i 1-10 -n 5”. Strace schrijft de uitvoer van het commando shuf naar stdout. De systeemaanroepen die het commando shuf aan het besturingssysteem heeft gericht worden naar stderr gestuurd. Herschijf de bovenstaande opdracht zodat nu uitvoer van shuf naar /dev/null wordt gestuurd en waarbij de systeemaanroepen kunnen worden overlopen door een pipe naar het commando less.

In [ ]:
```
%%bash
strace shuf -i 1-10 -n 5 1> /dev/null 2>&1 | less
```

25. Open de manpagina van de opdracht tr. Maak ook een bestand aan met uitsluitend leestekens en kleine letters. Hoe kan je met tr alle tekst uit dit bestand omzetten naar een hoofdletters? Het commando tr kent geen bestandparameters waardoor je met input en output redirection zal moeten werken. Bemerk ook dat het een slecht idee is om binnen een proces van een bestand te lezen en er ook naartoe te schrijven. Maak dus gebruik van een tijdelijk bestand dat je na tr met de opdracht cp naar het originele bestand kopieert (en waardoor dus het oorspronkelijk bestand overschreven wordt).

In [ ]:
```
%%bash
echo "wowie ik ben een beetje... moe van de hectische examenweek, maar ik ben er no
tr '[:lower:]' '[:upper:]' < bestand.txt > bestand2.txt
```

```
cp bestand2.txt bestand.txt
rm bestand2.txt
```

26. Hoe wordt het regeleinde aangegeven in Windows en hoe wordt dit gedaan in Linux? Het gebruik van het Linux-formaat in Windows geeft problemen, dewelke? Het gebruik van het DOS-formaat geeft dan weer nog grotere problemen in Linux. Waarom?

- In Windows: CR LF
- In Linux: LF
- Windows-programma's die alleen CRLF herkennen kunnen moeite hebben met LF-geformatteerde bestanden, waardoor deze onjuist worden weergegeven of verwerkt.
- Linux-programma's en -scripts kunnen problemen hebben met de extra \r-karakters van CRLF-geformatteerde bestanden, wat leidt tot fouten bij script-uitvoering en inconsistent gedrag van tekstverwerkingstools.

27. Gebruik het commando od om de inhoud van het bestand /etc/passwd hexadecimaal, in groepjes van 1 byte, op het scherm te tonen. Doe hetzelfde met het commando xxd en vergelijk de uitvoer van beide commando's. Bekijk uitvoerig de manpagina's van beide opdrachten!

In [ ]:
```
%%bash
od -t x1 /etc/passwd | head -n 5
xxd -g 1 /etc/passwd | head -n 5
```

28. Bij vraag 12 werd er gevraagd om 4 bytes (32 bits) te lezen van /dev/random. Zet die uitvoer met het commando od nu om naar een decimaal getal. Hoe zet je het om naar een strikt positief getal? (unsigned dus)

In [ ]:
```
%%bash
od -An -tu4 <(head -c 4 /dev/random)
```

29. Gebruik het commando cut (zie manpage) om alle gebruikersnamen (1e veld) uit het wachtwoordenbestand te halen?

In [ ]:
```
%%bash
cut -d: -f1 /etc/passwd
```

30. Hoe kan je het bestand /etc/passwd sorteren op het eerste veld? Zie de manpage van sort.

In [ ]:
```
%%bash
sort -t: -k1,1 -n /etc/passwd
```