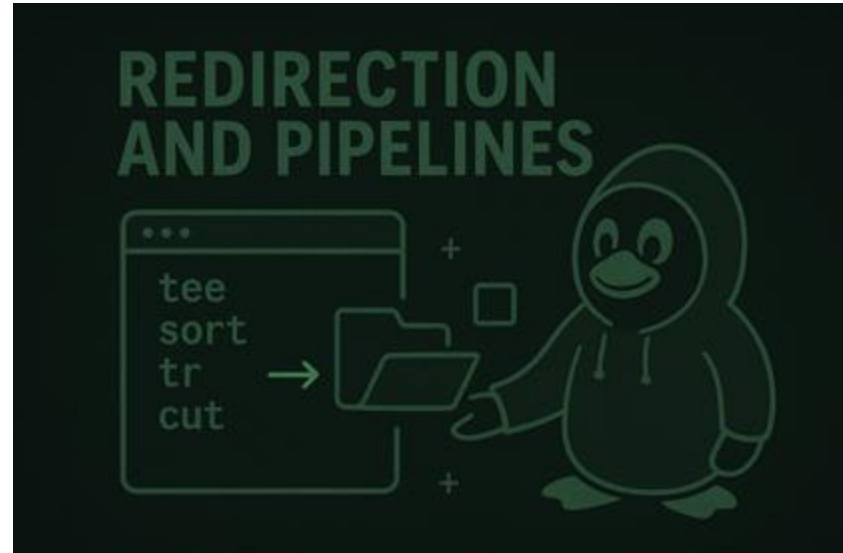


# Operating System fundamentals

Redirection en pipelines



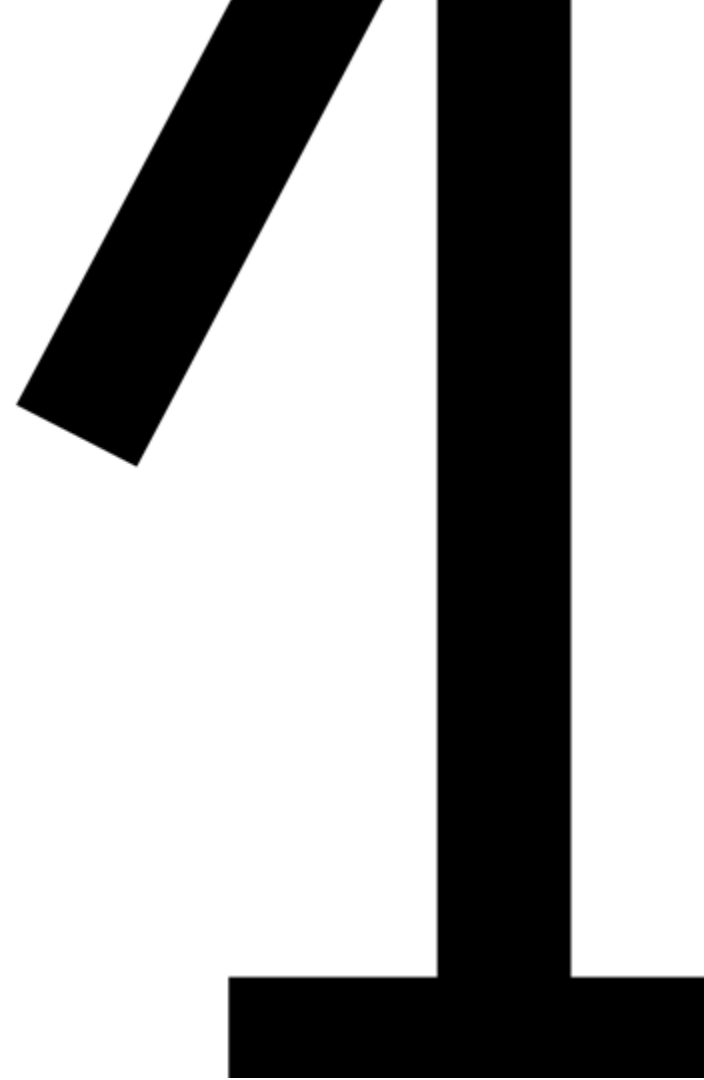
# Inhoud

1. Standaard streams
2. Redirection
3. Pipelining
4. Extra commando's

# Cursus

- chapter 5: Create, View and Edit Text Files





---

# Standaard streams

# Standaard streams

- ieder programma/commando kan
  - lezen van standaard input (**stdin**)
    - in Java is dit System.in
  - schrijven naar standaard output (**stdout**)
    - in Java is dit System.out
  - schrijven naar standaard error (**stderr**)
    - in Java is dit System.err

# Voorbeeld in Java

```
import java.util.Scanner;
public class MijnCommando {
    public static void main(String args[]) {
        System.out.print("Geef een woord: ");
        String s = new Scanner(System.in).nextLine();
        if (s.isEmpty()) {
            System.err.println("Fout: Je gaf een lege string in");
        } else {
            System.out.println("Je gaf de string '" + s + "' in");
        }
    }
}
```

# Voorbeeld: sort

- **sort** leest lijnen uit stdin, sorteert deze en schrijft de gesorteerde waarden naar stdout

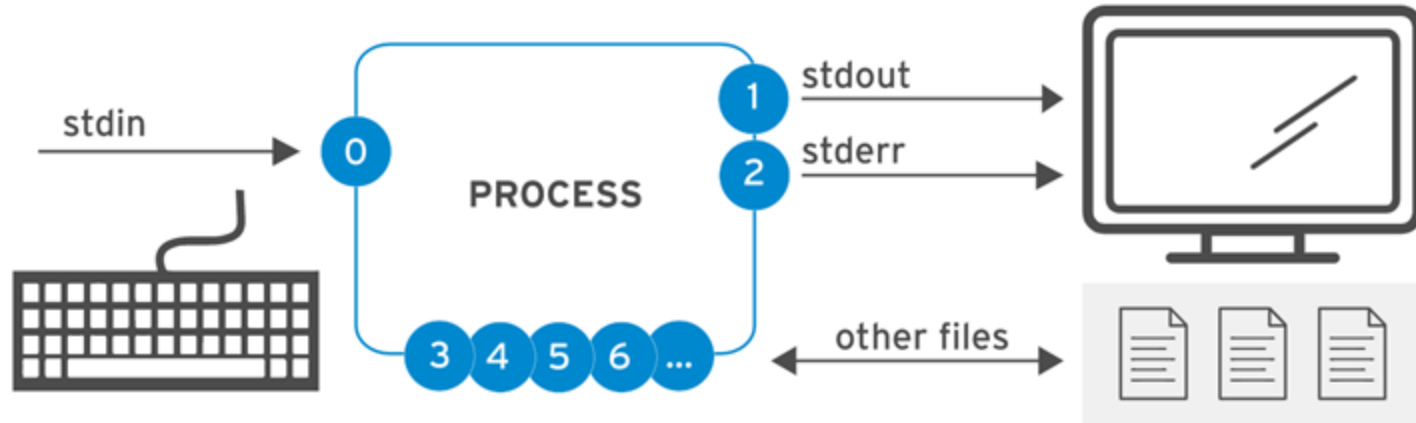
- voorbeeld:

- **sort** <enter>
- harry <enter>
- sally <enter>
- saisy <enter>
- sarrry <enter>
- <ctrl+d>

Voorbeeld:

- maak bestand filename.txt en zet er 3 regels tekst in
- sort filename.txt

# Standaard streams

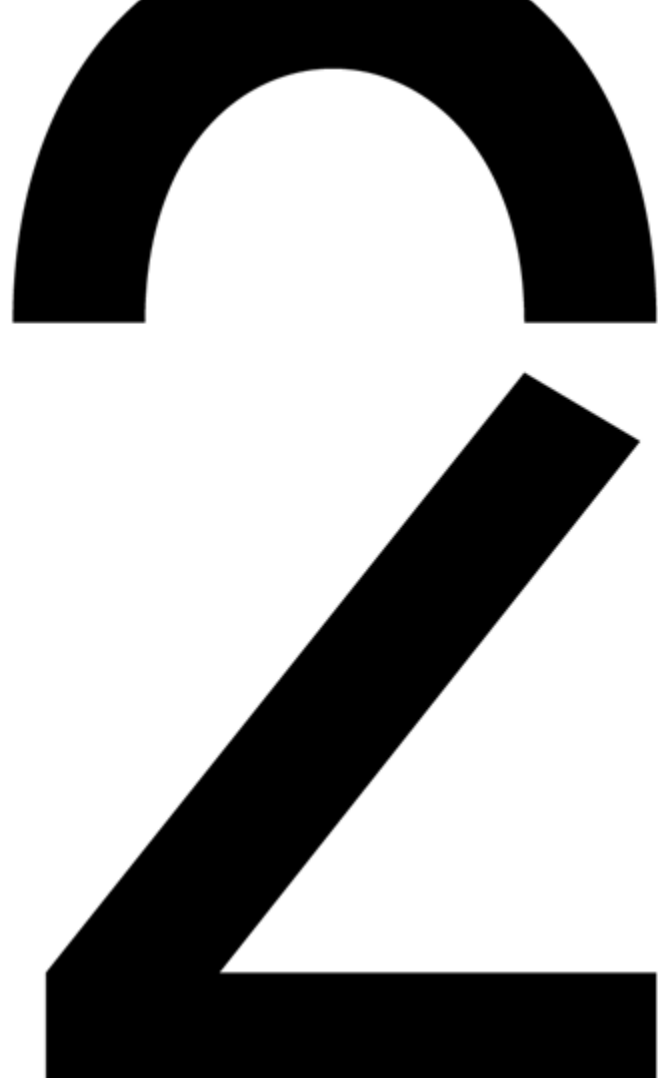


Number	Channel name	Description	Default connection	Usage
0	<code>stdin</code>	Standard input	Keyboard	read only
1	<code>stdout</code>	Standard output	Terminal	write only
2	<code>stderr</code>	Standard error	Terminal	write only
3+	<i>filename</i>	Other files	none	read, write, or both



---

# Redirection



# Redirection

- Je kan standaard input/output/error van en naar een bestand "redirecten"
  - **sort** <inputFile.txt
  - **sort** <inputFile.txt >outputFile.txt
  - **sort** <inputFile.txt >>outputFile.txt
  - **sort** <inputFile.txt >outputFile.txt **2>** fouten.txt
  - **sort** <inputFile.txt >outputFile.txt **2>** /dev/null
  - **sort** <inputFile.txt **&>**alleOutput.txt

# Redirection

`find /etc -name passwd`

→ *resultaat*

`/etc/passwd`

`/etc/pam.d/passwd`

→ Fouten

`find: '/etc/cups/ssl': Permission denied`

...

> Redirect STDOUT naar bestand

>> Redirect STDOUT en **voeg toe** aan bestand

2> Redirect STDERR naar bestand (`2>/dev/null` = gooi weg)

2>> Redirect STDERR en **voeg toe** aan bestand

&> Redirect zowel STDOUT als STDERR naar bestand - equivalent aan `>file 2>&1`

&>> Redirect zowel STDOUT als STDERR en **voeg toe** aan bestand

# Redirection

```
find /etc -name passwd >output.txt 2>>error.log
```

- STDOUT wordt bewaard in output.txt  
STDERR wordt toegevoegd aan error.log

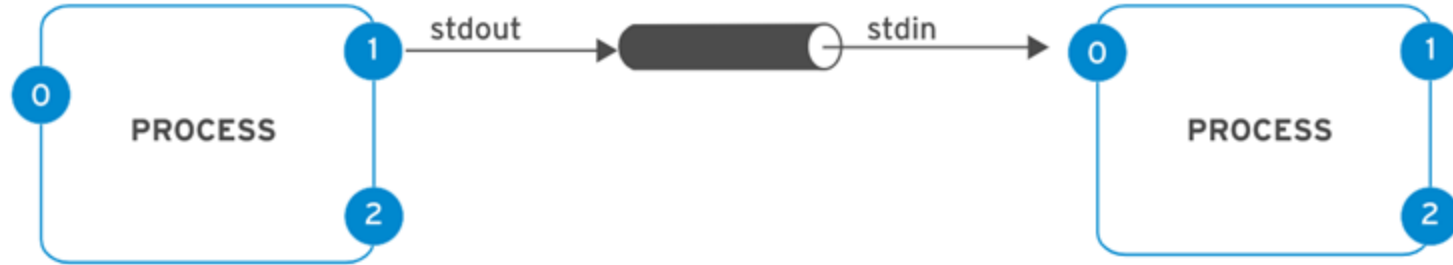
```
find /etc -name passwd &> /tmp/all-message-output
```

- foutmeldingen en output worden beiden bewaard in het bestand "all-message-output" in de /tmp directory

---

# Pipelining

# Pipelining



- Commando's combineren door stdout door te geven aan stdin van een ander commando

# Pipeline voorbeelden

```
$ ls | sort -r
```

```
$ ls | wc
```

```
$ ls -l /usr/bin | less
```

```
$ fortune | cowsay
```

```
$ cat /etc/group | nl | sort
```

```
$ ls -t | head -n 10 > /tmp/first-ten-changed-files
```

```
$ cat /etc/passwd | sort | less
```

# Tee gebruiken in een pipeline

Bekijk dit commando:

```
$ ls > /tmp/saved-output | less
```

Er verschijnt geen output op het scherm...

→ Gebruik het "tee" commando om een "T" in de pipeline te creëren

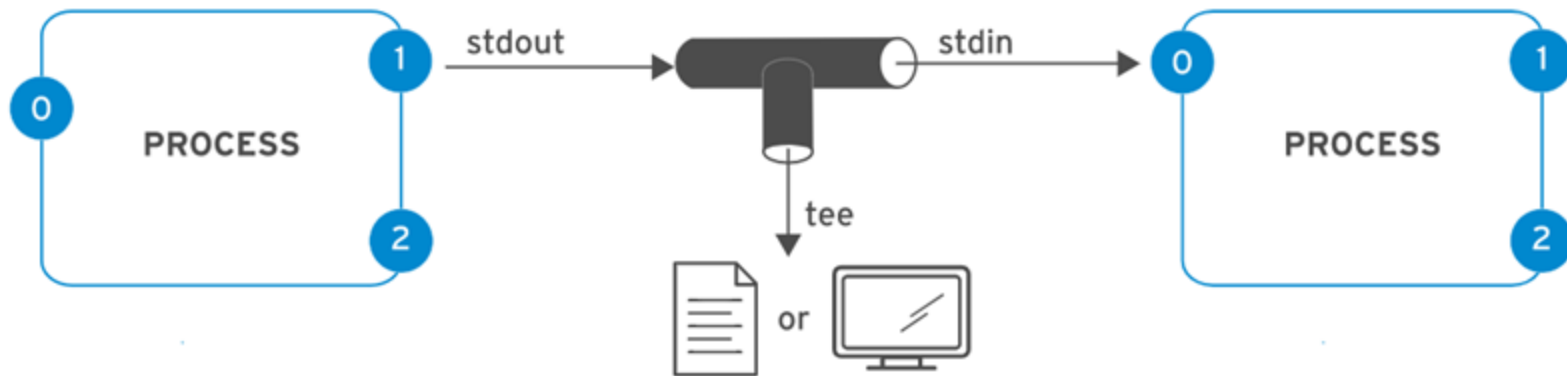


Figure 5.9: Process I/O piping with tee



# Voorbeelden met tee

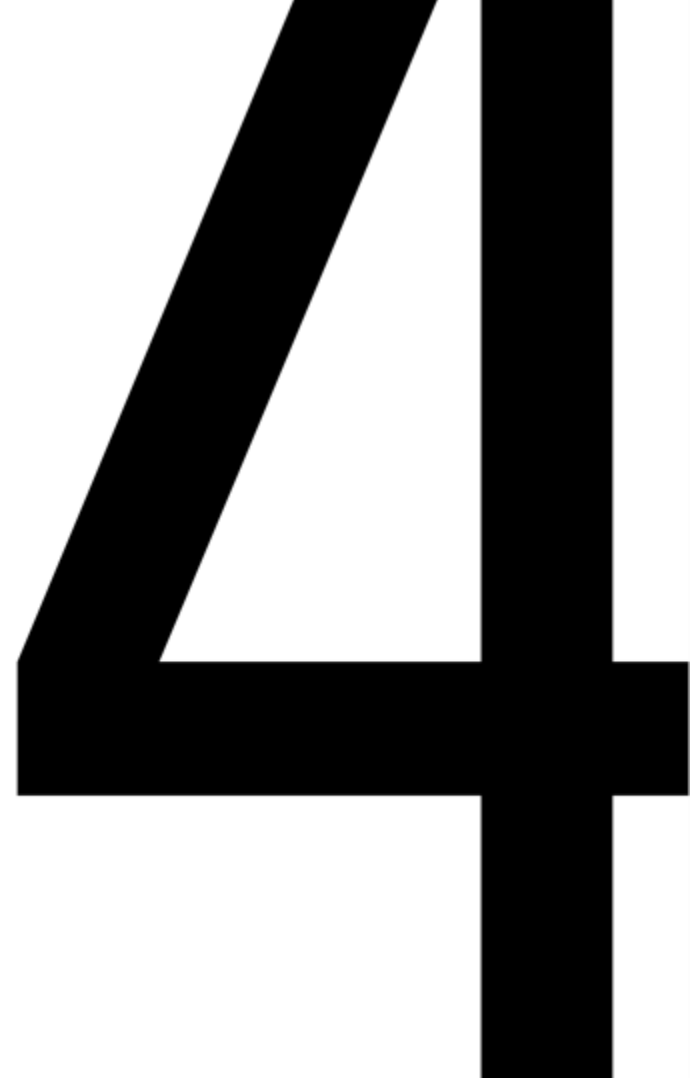
```
$ ls | tee /tmp/saved-output | less
```

Bewaar output in /tmp/saved-output

Toon output in less

```
$ ls -l | tee -a /tmp/append-files
```

**-a** option zorgt dat tee de output toevoegt aan het bestand



---

## **Extra commando's**

# Veel voorkomende commando's in pipelines

Volgende commando's zijn dikwijls handig in pipelines:

- head, tail, wc (zie ook vroeger)
- sort
- uniq
- tr
- cut

Oefening:

Hoe selecteer je lijnen 5 tot en met 10 uit een bestand?

# Sorteren met sort en uniq

- sort
  - **sort -f** -> case insensitive (is default!)
  - **sort -n** -> numerisch ipv alfabetisch
  - **sort -u** -> elimineer dubbele lijnen
  - **sort -r** -> omgekeerd sorteren
- uniq: elimineer dubbele lijnen
  - lijnen moeten na elkaar staan! (dus best nà sort)
  - cat tekst.txt | sort | **uniq**
  - cat tekst.txt | **sort -u**

# Kolommen selecteren met cut

- cut: selecteer een kolom uit standard input
  - `cat /etc/passwd | cut -d ":" -f 4`
  - “delimiter” is standaard tab
  - `ls -l | cut -d " " -f 3`
- handig bij “CSV bestanden”
  - wat is een CSV bestand?
  - `cat bestand.csv | cut -d "," -f 1,2`

# Oefening

1. Filter uit het bestand `/etc/group` de eerste kolom met namen.
2. Sorteer die kolom alfabetisch, zet er lijnnummers voor en schrijf het resultaat weg in het bestand `/tmp/groupnames`

# Translate (tr)

Gebruik "tr" om karakters te vervangen door andere:

- `cat file.csv | tr ',' ';'`
- `echo "GroTe En kleine LEtteRS" | tr "a-z" "A-Z"`

Je kan er ook karakters mee verwijderen

- `echo "DIT EN DAT" | tr -d 'A'`

Je kan er ook achtereenvolgende karakters mee vervangen door 1 ("squeeze")

- `ls -las | tr -s " " | cut -d " " -f 5`

---

# Oefeningen



# Oefeningen

- KdG
  - ex0501
- RedHat
  - ch05s02

