

# FIFO's

Veerle Ongenae

# FIFO's

- Basisconcepten
- FIFO's in C
- FIFO-opdrachten

# Basisconcepten

- First In First Out
- named pipe
- verschil met pijp :
  - device in bestandssysteem
  - processen hoeven geen gemeenschappelijke voorouder te hebben
  - fifo's blijven in bestandssysteem na gebruik

# FIFO's

- Basisconcepten
- FIFO's in C
- FIFO-opdrachten

# FIFO's In C

```
#include<sys/types.h>
```

```
#include <sys/stat.h>
```

```
int mkfifo(const char *path, mode_t mode)
```

*return* : 0 bij succes, -1 en errno bij fout

# FIFO's in C

- path
  - Naam FIFO
- mode
  - permissies

# FIFO modes

| <b>Constante</b> | <b>Omschrijving</b> |
|------------------|---------------------|
| S_IRUSR          | user read           |
| S_IWUSR          | user write          |
| S_IRGRP          | group read          |
| S_IWGRP          | group write         |
| S_IROTH          | other read          |
| S_IWOTH          | other write         |

```
mkfifo("/tmp/m_fif", S_IRUSR | S_IWUSR)
```

# FIFO's

- Basisconcepten
- FIFO's in C
- FIFO-opdrachten

# Voorbeeld

- FIFOServer.cpp
- FIFOClient.cpp

# FIFO's blokkeren

- schrijven of lezen naar een fifo is een blokkerende actie
  - Proces wacht tot een ander proces de fifo opent om te lezen of te schrijven
- Oplossing
  - gebruik non-blocking descriptor
    - optie `O_NONBLOCK`