

**I2181-B**  
**LINUX**  
**APPELS SYSTÈME**

**SELECT - POLL**

# INTRODUCTION

- Certains appels systèmes sont bloquants
  - Ex: read
- Ces appels bloquants peuvent être gênants
  - Imaginez un serveur de jeu devant lire la réponse de plusieurs joueurs
  - Si un joueur ne lui envoie pas sa réponse, le serveur ne fera rien et restera bloqué en attendant la réponse
  - Le serveur sera également obligé de traiter les joueurs dans un ordre bien précis, impossible de traiter les réponses des joueurs de manière asynchrone

# INTRODUCTION

- Il existe 2 appels systèmes permettant de :
  - Surveiller des **descripteurs de fichiers**  
(fd, sockfd, ...)
  - Transformer un appel système sur descripteur de fichier bloquant en non bloquant.



# SELECT - POLL

- SELECT est l'appel système le plus ancien permettant de transformer des appels bloquants en non bloquants.
  - Nous le citons simplement dans ce cours car on le rencontre encore beaucoup dans la littérature et programmes mais nous utiliserons plutôt POLL.
- POLL est un appel système plus récent visant à remplacer SELECT
- Ces 2 appels systèmes ont un fonctionnement similaire.

# POLL

```
#include <poll.h>
```

```
int poll(struct pollfd *fds, int nfds,  
         int timeout);
```

- **fds** : un tableau de *struct pollfd* contenant des file descriptors à surveiller
- **nfds**: le nombre de file descriptors présents dans **fds**
- **timeout** : le temps que poll doit surveiller les fds
  - Ex: 0 → surveillance instantanée
  - Ex: 10 → poll surveille les fds pendant 10 secondes. Ceci est bloquant !

# POLL

```
#include <poll.h>
```

```
int poll(struct pollfd *fds, int nfds,  
         int timeout);
```

- Renvoie 0 si le *timeout* a expiré.
- Renvoie le nombre de structures ayant le champ *revents* non nul
- Renvoie -1 en cas d'erreur.



# POLL

- Chaque élément du tableau `fds` contient la structure ci-dessous

```
struct pollfd {  
    int fd;           /* file descriptor */  
    short events;     /* requested events */  
    short revents;    /* returned events */  
};
```

- `fd` : le numéro du descripteur de fichier
- `events` : paramètre pour indiquer à l'appel système *poll* le type de surveillance:
  - `POLLIN` : surveillance en lecture
  - `POLLOUT` : surveillance en écriture
- `revents` : paramètre rempli par l'appel système `POLL`
  - `POLLIN` : il y a des données à lire
  - `POLLOUT` : il y a des données à écrire

# EXAMPLE

Demo

Exemple\exemple18\_sans\_poll VS

Exemple\exemple19\_avec\_poll