

# Projet systèmes d'exploitation

## shm

```
// Retourne un descripteur de flux sur un segment de mémoire partagé
// Les paramètres sont les mêmes que la fonction 'shm_open'
streamd_t* _shm_open(const char* name, int oflag, mode_t mode) ;

// Ferme le segment de mémoire partagé
int _shm_close(streamd_t* sd) ;

// Lit un maximum de size octets dans sd et les stockent dans buffer
// Retourne le nombre d'octets lues, -1 si erreur
// la fonction est bloquante
int _shm_read(streamd_t* sd, char* buffer, size_t size) ;

// Ecrit size octets de buffer dans sd
// Retourne le nombre d'octets écrits
inr _shm_write(streamd_t* sd, char* buffer, size_t size) ;
```

## file de messages

```
// Retourne un descripteur sur une file de messages
// Les paramètres sont les mêmes que la fonction 'mq_open'
streamd_t* _mq_open(const char* name, int oflag, mode_t mode) ;

// Ferme la file de messages
int _mq_close(streamd_t* sd) ;

// Lit un maximum de size octets dans sd et les stockent dans buffer
// Retourne le nombre d'octets lues
int _mq_read(streamd_t* sd, char* buffer, size_t size) ;

// Ecrit size octets de buffer dans fd
// Retourne le nombre d'octets écrits
int _mq_write(streamd_t* fd, char* buffer, size_t size) ;
```

## FIFO

```
// Retourne un descripteur sur un tube nommé
streamd_t* _fifo_open(const char* name, int oflag, mode_t mode) ;

// Ferme le tube nommé
int _fifo_close(streamd_t * sd) ;

// Lit un maximum de size octets dans fd et les stockent dans buffer
// Retourne le nombre d'octets lues
```

```

int _fifo_read(streamd_t* fd, char* buffer, size_t size) ;

// Ecrit size octets de buffer dans fd
// Retourne le nombre d'octets écrits
int _fifo_write(streamd_t* fd, char* buffer, size_t size) ;

```

## STREAM

```

// Retourne
stream_t* stream_init() ;

// Ouvre le descripteur
void stream_open(stream_t* op, const char* name, int oflag, mode_t mode) ;

// Ferme le flux de données
int stream_close(stream_t* op) ;

// Lit un maximum de size octets dans le flux et les stockent dans buffer
// Retourne le nombre d'octets lues
int stream_read(stream_t* op, char* buffer, size_t size) ;

// Ecrit size octets de buffer dans le flux
// Retourne le nombre d'octets écrits
int stream_write(stream_t* op, char* buffer, size_t size) ;

struct {
    streamd_t* (_open) (const char*, int , mode_t) ;
    int (_close) (streamd_t* fd) ;
    int (_read) (streamd_t*, char*, size_t) ;
    int (_write) (streamd_t*, char*, size_t) ;
    descripteur* fd ;
} stream_t ;

```

```

Struct {
    union {
        int fd ;
        mqd_t mq ;
    } sd;
} streamd_t ;

```

## Algorithme du server

```
Initialisation(op)
ouverture(op)
tant que lire(op, buffer) > 0 faire
    traitement par un processus fils
ftantque
fermer(op)
```