

SELECT()

Héctor Menéndez

AIDA Research Group
Computer Science Department
Universidad Autónoma de Madrid

30 de enero de 2013

Index

1 La función `select()`

La estructura select()

Listing 1: Definición de la estructura select()

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

int select(int nfds, fd_set *readfds, fd_set *writefds,
           fd_set *exceptfds, struct timeval *timeout);

void FD_CLR(int fd, fd_set *set);
int  FD_ISSET(int fd, fd_set *set);
void FD_SET(int fd, fd_set *set);
void FD_ZERO(fd_set *set);
```

La estructura FD_CLR()

Listing 2: Definición de la estructura FD_CLR()

```
void FD_CLR(int fd , fd_set *set );
```

- Quita un descriptor al conjunto.

La estructura FD_ISSET()

Listing 3: Definición de la estructura FD_ISSET()

```
int FD_ISSET(int fd , fd_set *set );
```

- Nos indica si ha habido algo en el descriptor fd dentro de set. Cuando select() sale, debemos ir interrogando a todos los descriptors uno por uno con esta macro.

La estructura FD_SET()

Listing 4: Definición de la estructura FD_SET()

```
void FD_SET(int fd , fd_set *set);
```

- Añade un descriptor al conjunto.

La estructura FD_ZERO()

Listing 5: Definición de la estructura FD_ZERO()

```
void FD_ZERO( fd_set *set );
```

- Limpia un conjunto.

La estructura select()

Listing 6: Definición de la estructura select()

```
int select(int nfd, fd_set *readfds, fd_set *writefds,  
          fd_set *exceptfds, struct timeval *timeout);
```

- Se queda bloqueada hasta que ocurre un evento en alguno de los conjuntos. Después se analiza qué descriptor ha sido modificado y se vuelve a consultar para esperar.
- Se puede establecer un timeout.