Escucho y olvido, veo y recuerdo,
hago y entiendo. *

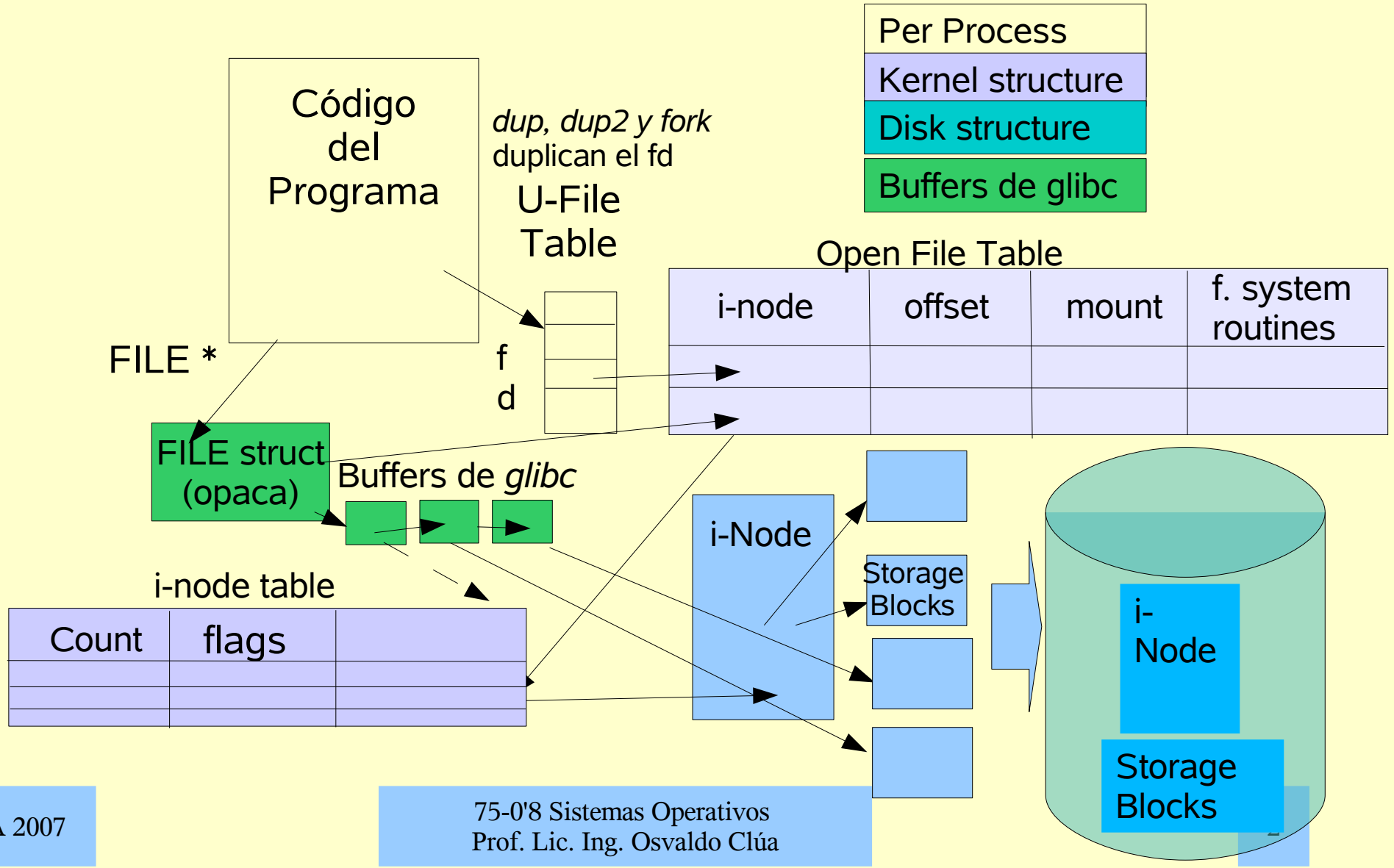
75-08 Sistemas Operativos
Lic. Ing. Osvaldo Clúa
2007

Facultad de Ingeniería
Universidad de Buenos Aires

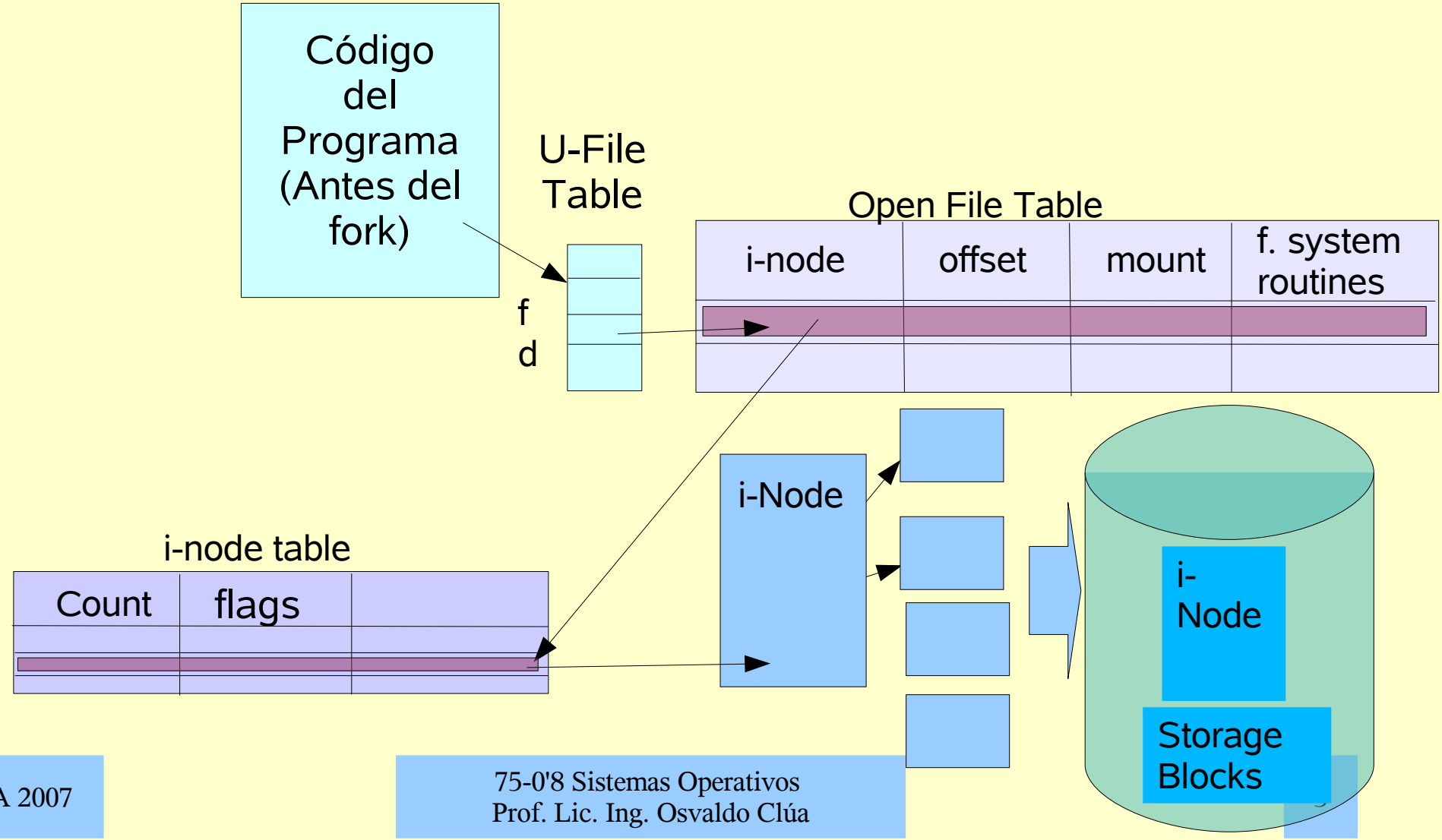
Lab 3 – Files

* Proverbio chino atribuido a Confucio (2000 AC).

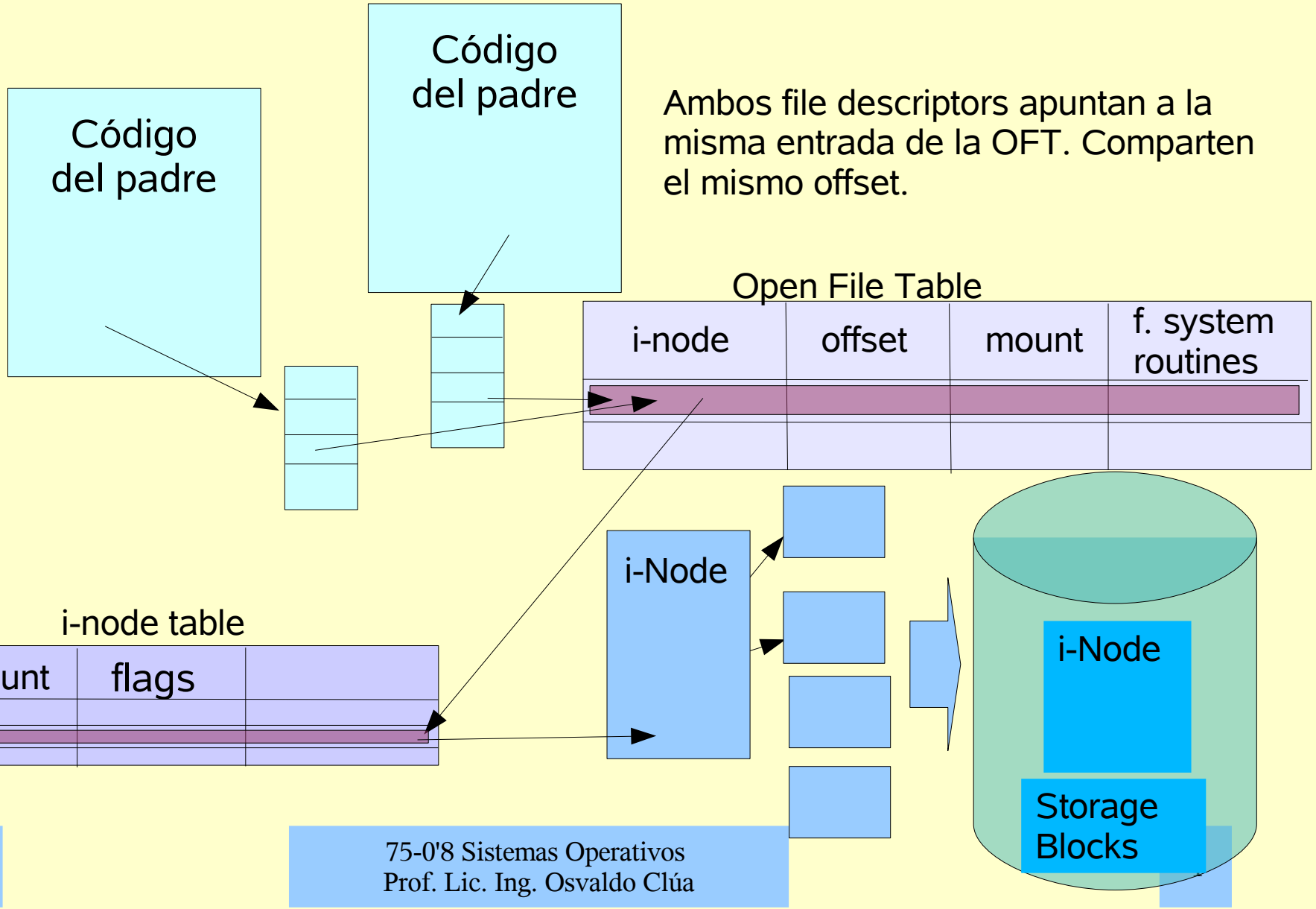
Estructuras usadas en el procesamiento de Archivos



0fork.cc


Código del Programa (Antes del fork)
U-File Table
f
d
Open File Table
i-node
offset
mount
f. system routines
i-Node
i-node table
Count
flags
i-Node
Storage Blocks

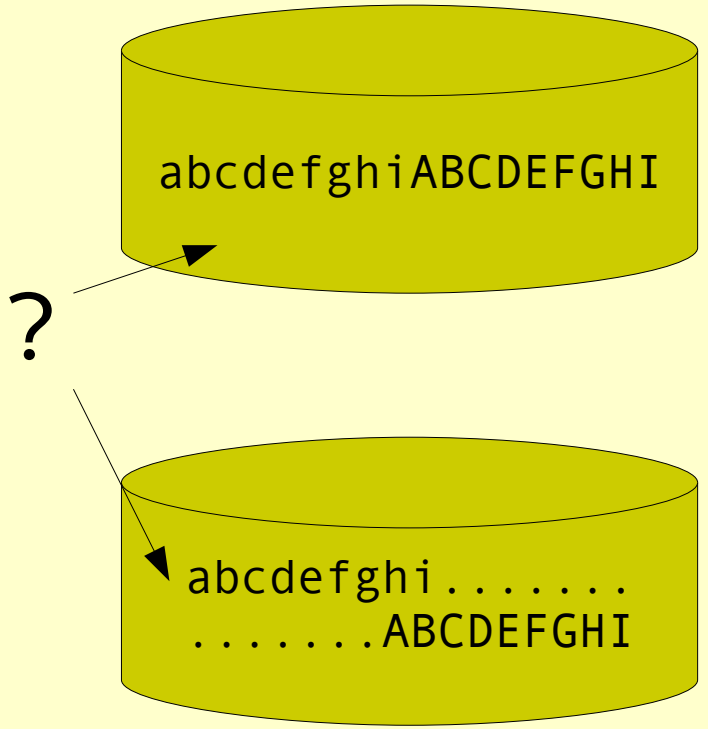
0fork.cc
(durante la existencia del hijo)



hole.cc

El siguiente trozo de código crea un archivo y escribe dos *strings*, saltando algunos bytes

```
string uno ("abcdefghij");
string dos ("ABCDEFGHIJ");
.....
out=creat (fout.c_str(),00666);
  cant=write(out,uno.c_str(),uno.length());
  lseek (out,40,SEEK_SET);
  cant=write(out,dos.c_str(),dos.length());
  close (out);
```



leedir.cc

Lee la estructura de un directorio usando las primitivas de directorios.

```
struct dirent {
  ino_t          d_ino;       /* inode number */
  off_t          d_off;       /* offset to the next dirent */
  unsigned short d_reclen;    /* length of this record */
  unsigned char  d_type;      /* type of file */
  char           d_name[256]; /* filename */
;
```

flock

Comprobar el funcionamiento del código ejemplo y su interacción con el comando.

flock (1) - acquire a file lock and then execute a command with the lock held

flock [--shared | --timeout=seconds] lockfile command ..

flock(2) - apply or remove an advisory lock on an open file

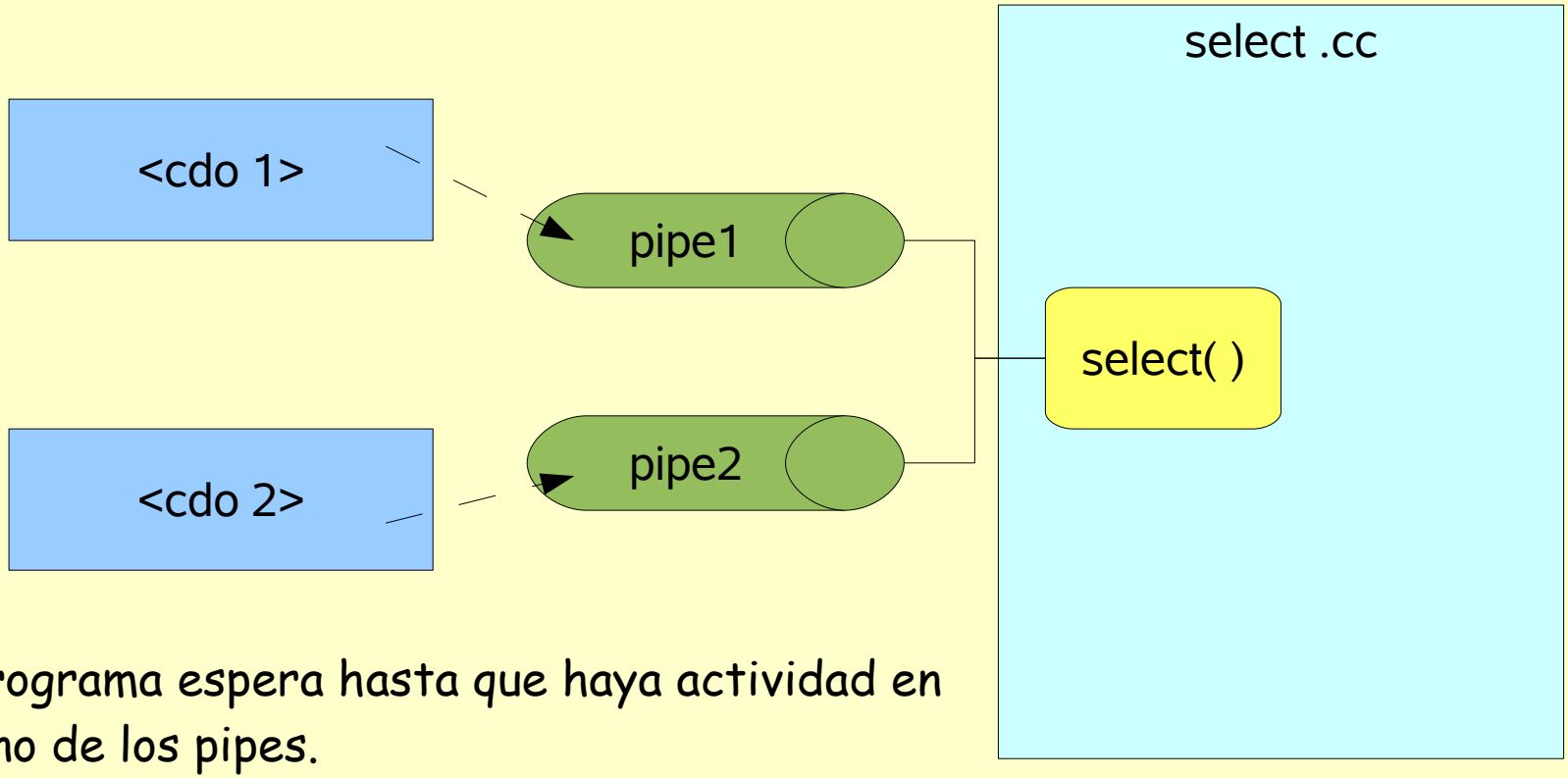
```
#include <sys/file.h>
 int flock(int fd, int operation);
```

select

Espera por actividad en un *fd*

```
synchronous I/O multiplexing
 /* According to POSIX.1-2001 */
#include <sys/select.h>
/* According to earlier standards */
#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);
```

named pipes (select.cc)



El programa espera hasta que haya actividad en alguno de los pipes.

¿Porque se hace el open del pipe con la opción NOBLOCK?

```
int p2= open(argv[2],O_NONBLOCK|0600);
```

mmap

mmap, munmap - map or unmap files or devices into memory

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

void *mmap(void *start, size_t length, int prot, int flags, int fd, off_t offset);
int munmap(void *start, size_t length);
```

Los programas permiten comparar performance (usando *strace -cT* ver Lab 1) entre acceso común y acceso mapeado a memoria.