

Llamadas al Sistema Unix

Sistemas Operativos

Escuela de Ingeniería Civil Informática

Entrada/Salida sin bloqueo (o no
bloqueante)

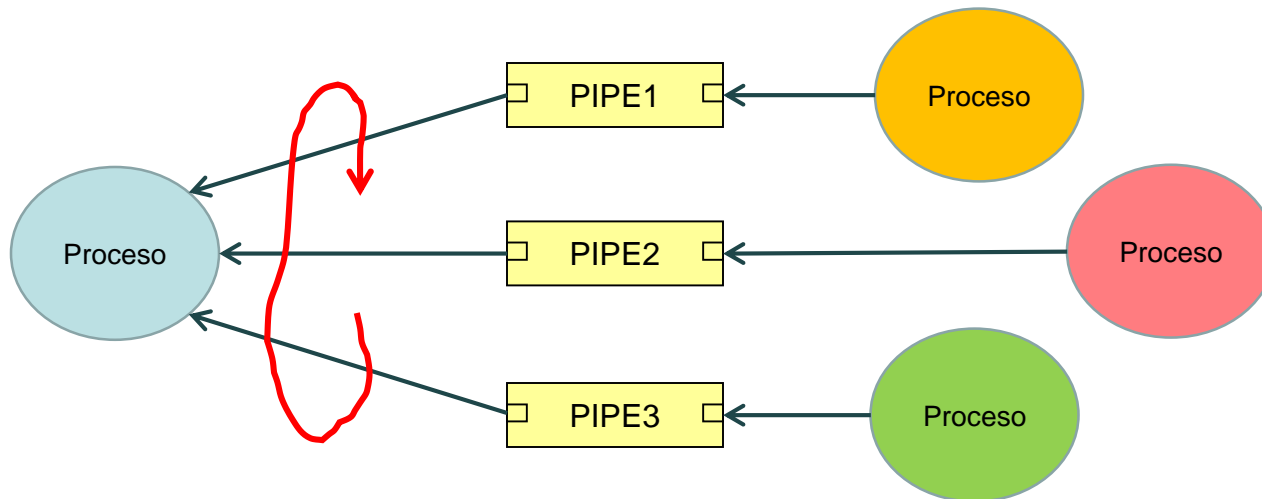
- `select()`



UNIVERSIDAD DEL BÍO-BÍO

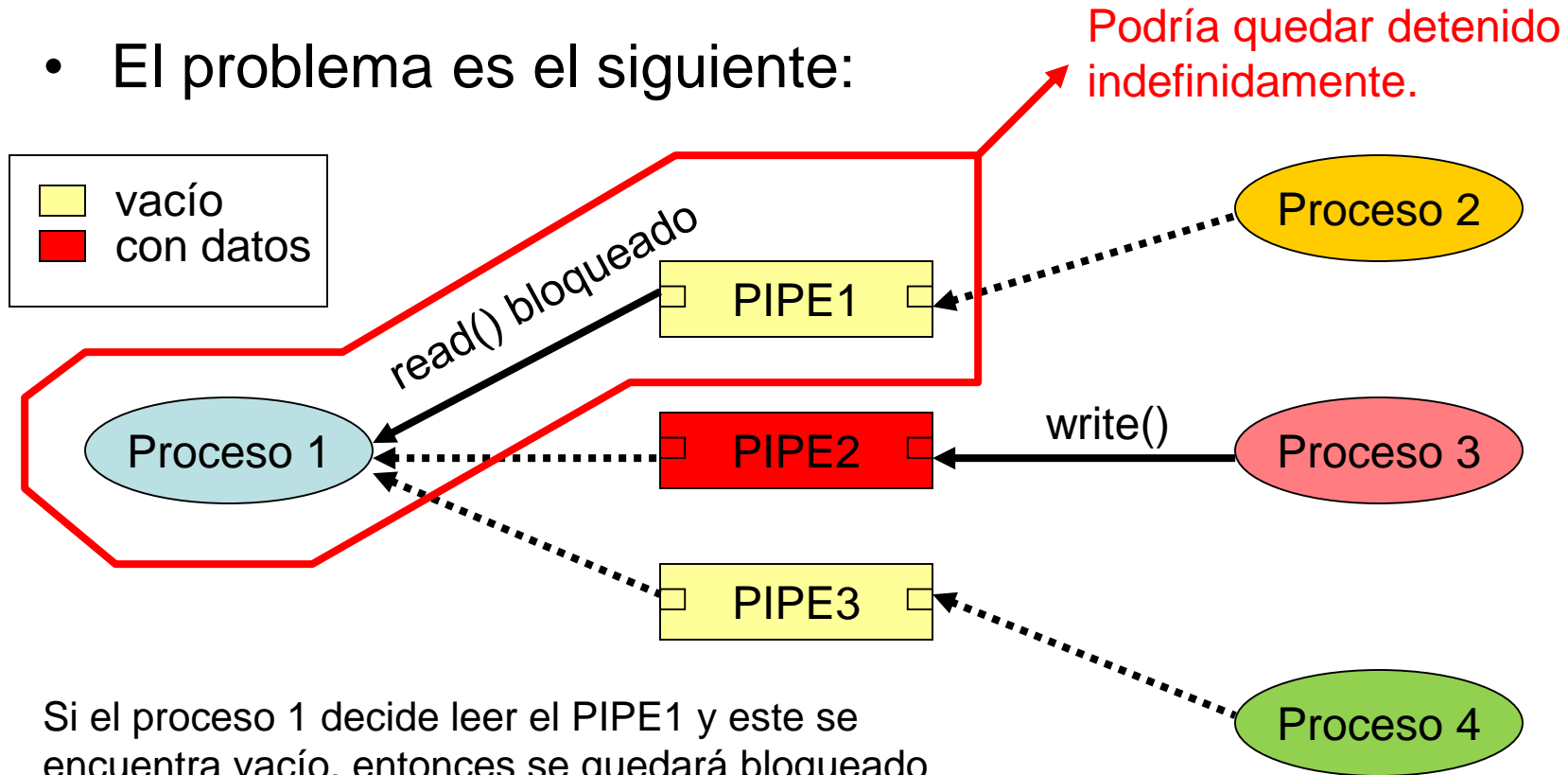
El problema

- En ocasiones un descriptor no se encuentra listo para una operación:
 - Lectura – `read(...)` el descriptor puede estar vacío
 - Escritura – `write(...)` el descriptor puede estar lleno
 - En cualquiera de estos casos el proceso **se bloquea**.
- Si un proceso debe vigilar varios buffers simultáneamente, y alguno se bloquea, ¿cómo puede revisar el resto?



E/S no bloqueante

- El problema es el siguiente:



Si el proceso 1 decide leer el PIPE1 y este se encuentra vacío, entonces se quedará bloqueado (a la espera) de que lleguen datos para poder realizar la lectura.

Si estando bloqueado, el proceso 3 envía datos al PIPE2, este podría llenarse dejando bloqueado al proceso 3, ya que el proceso 1 se encuentra bloqueado esperando datos en el PIPE1. Lo mismo podría suceder con el proceso 4.

E/S Multiplexada

- La llamada al sistema *select* provee una forma de vigilar varios descriptores a la vez sin necesidad de consultar por todos dentro de un ciclo (como se hace en el polling).
- Se deja libre al proceso (que no polling)
- Se notifica al proceso cuando los descriptores que desea manipular estén listos para lectura o escritura.

Prototipo de *select*

Archivo cabecera	<pre>#include <sys/types.h> #include <sys/sys.h></pre>		
Formato	<pre>int select(int nfd, fd_set *readfds, fd_set *writefds, fd_set *exceptfds, struct timeval *timeout);</pre>		
Salida	Exito	Fallo	Valor en errno
	Número de descriptores listos	-1	Si

- ***nfd*** es el número del descriptor mayor a vigilar más uno (+1).
- ***readfds*** es el conjunto de descriptores que se vigilarán por una entrada.
- ***writefds*** es el conjunto de descriptores que se vigilarán por una escritura.
- ***exceptfds*** es el conjunto de descriptores que se vigilarán por una situación excepcional.
- ***timeout*** es el tiempo que select esperará antes de retornar. Si es NULL el bloqueo será indefinido, hasta que existan descriptores listos.

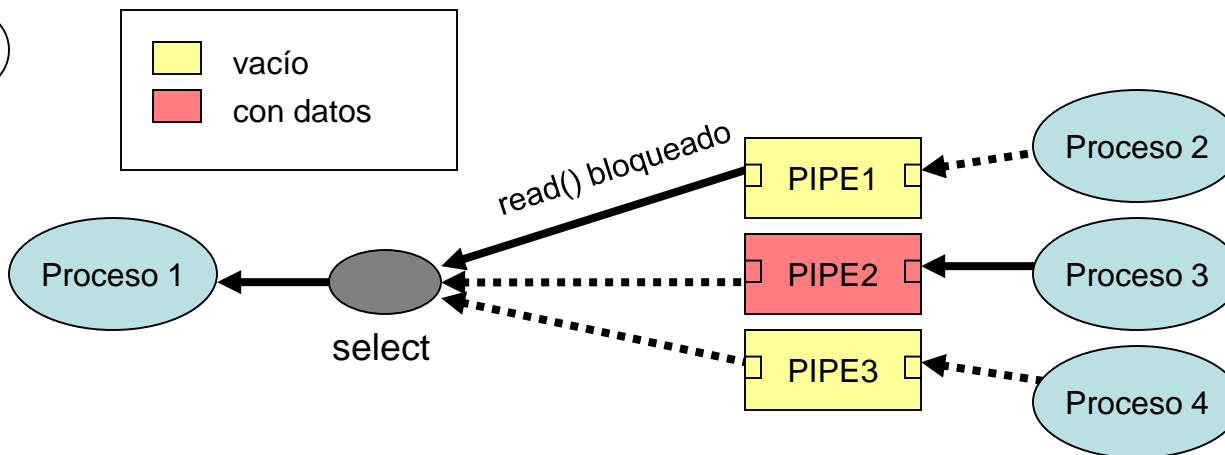
Funcionamiento de *Select*

1 ...
select (nfds, readfds, writefds, exceptfds, timeout);
...

Agregar los descriptors por los cuales se necesita consultar su estado a los conjuntos según se corresponda.

readfds	writefds	exceptfds
PIPE1 PIPE2 PIPE3	vacío	vacío

2



Revisar los conjuntos para saber qué descriptors están listos

readfds	writefds	exceptfds
PIPE2	vacío	vacío

Administrar las listas de descriptores

- Para agregar un descriptor a cada conjunto se utilizan las siguientes macros:
 - `void FD_CLR(int fd, fd_set *set);`
Quita el descriptor *fd* del conjunto *set*.
 - `int FD_ISSET(int fd, fd_set *set);`
Pregunta si *fd* está en el conjunto *set*.
 - `void FD_SET(int fd, fd_set *set);`
Agrega el descriptor *fd* al conjunto *set*.
 - `void FD_ZERO(fd_set *set);`
Limpia el conjunto (asigna cero)

Consideraciones par *Select*

- El parámetro *nfds* puede ser ubicado mediante una función máximo, la cual retorne el mayor de los descriptores. Luego se le suma 1.

```
int pipe1, pipe2, maxfd;
...
/*crear los pipes*/
maxfd= pipe1;
if (pipe2>maxfd)
    maxfd=pipe2;
/*ya sabemos cual es el mayor*/
```


Consideraciones para *Select*

- El parámetro *timeout*, permite indicarle a *select* cuánto tiempo debe esperar por el estado de los descriptores antes de retornar (NULL indica espera ilimitada).
- Esta variable es de tipo *struct timeval*, sus campos son los siguientes:

```
struct timeval tv;
tv.tv_sec= 5; /*5 segundos*/
tv.tv_usec= 20; /*20 microsegundos*/
...
```

Existen 1.000 microsegundos en un milisegundo, y 1.000 milisegundos en un segundo. Así entonces, existen 1.000.000 de microsegundos en un segundo.

- Si hay descriptores listos antes se retorna de inmediato.
- Si el proceso es interrumpido por una señal mientras espera, también se retorna con código de error -1 y se establece el valor EINTER en la variable *errno*.

Ejemplo de Select

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

int main() {
    fd_set rfd;
    struct timeval tv;
    int valret;
    /*crear el conjunto*/
    FD_ZERO(&rfd); /*limpia el conjunto de los descriptors para lectura*/
    FD_SET(0, &rfd); /*agrega un descriptor a este conjunto*/
    /*Establecer el tiempo de espera para select*/
    tv.tv_sec=5; /*5 segundos*/
    tv.tv_usec=0; /*0 microsegundos*/
    valret= select(1, &rfd, NULL, NULL, &tv); /*0= stdin -> teclado*/
    /*en este momento el valor de valret contiene basura*/
    if (valret) /*si hay un descriptor en el conjunto (1)*/
        printf("Los datos ya están disponibles\n");
        /*FD_ISSET(0, &rfd) será también verdadero*/
    else
        printf("Ningún dato en 5 segundos.\n");
    return(0);
}
```