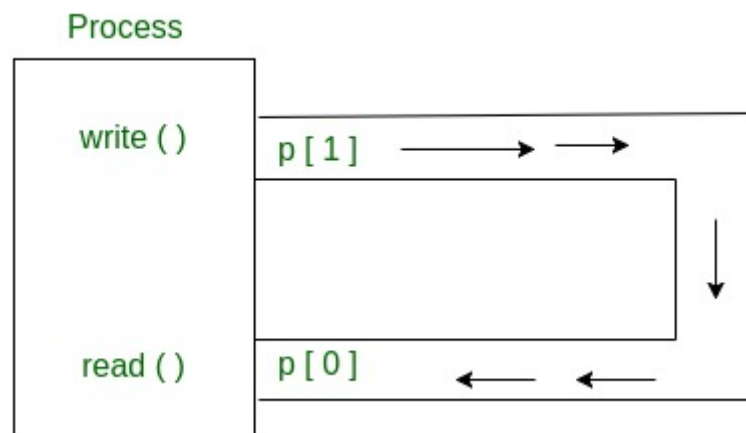

Sprawozdanie 6

Piotr Krześniak (303133)

1. Tworzenie prostych łączy jednokierunkowych

Pipe - jeden z mechanizmów komunikacji międzyprocesowej umożliwiający wymianę danych pomiędzy dwoma procesami. Odbywa się to najczęściej poprzez połączenie standardowego wyjścia jednego procesu ze standardowym wejściem drugiego. Potokiem w kodzie będzie dwuelementowa tablica deskryptorów, z czego zapis do pierwszego sprawi że będzie można odczytać te same dane z drugiego. Wielką zaletą potoku jest zapewnienie przez niego atomiczności zapisu i odczytu.

```
#include <unistd.h>
// Utwórz pipe, i zwróć odpowiednie deskryptory.
int pipe(
    int pfd[2] /* file descriptors */
);
/* Returns 0 on success or -1 on error (sets errno)
   pfd[0] przeznaczony jest do odczytu
   pfd[1] przeznaczony jest do zapisu
*/
```



Analiza read and write na podstawie manuala

I/O on pipes and FIFOs

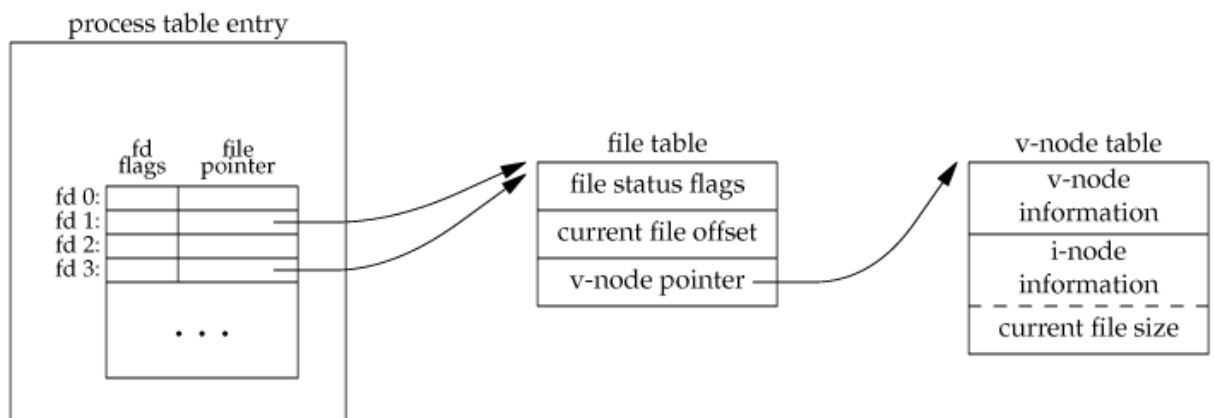
The only difference between pipes and FIFOs is the manner in which they are created and opened. Once these tasks have been accomplished, I/O on pipes and FIFOs has exactly the same semantics.

If a process attempts to read from an empty pipe, then `read(2)` will block until data is available. If a process attempts to write to a full pipe (see below), then `write(2)` blocks until sufficient data has been read from the pipe to allow the write to complete. Nonblocking I/O is possible by using the `fcntl(2)` `F_SETFL` operation to enable the `O_NONBLOCK` open file status flag.

Podsumowując, krytyczne sytuacje są zapobiegane przez wbudowane mechanizmy w przypadku czytania z pustego pipa oraz do zapisywania gdy jest zapełniony cały pipe.

2. Praca z łączami komunikacyjnymi

`dup` i `dup2`. Obie duplikują deskryptor pliku, z tym że pierwsza dla kopii wybiera pierwszy wolny deskryptor i przydziela mu numer, a druga pozwala nadać nam własną wartość deskryptora. Zduplikowane deskryptory wciąż wskazują w to samo miejsce (na tego samego pipe'a), mają tylko inną wartość. Ponadto operacje zamykania i otwierania deskryptora w funkcji jest atomiczna.



Kernel data structures after dup 1