

# YAUHENIYA PADBIAROSKAYA

308295

## Lab 6

### 1. Tworzenie prostych łączy jednokierunkowych:

1. `pipe(2)` – tworzy parę sprzężonych deskryptorów pliku potoku , wskazujących na i-węzeł potoku i umieszcza je w tablicy `fds[]`. `fds[0]` jest dla odczytu, a `fds[1]` dla zapisu.
2. `close()` – zamyka deskryptor pliku, już nie odnosi się do żadnego pliku i może być użyty ponownie.

`fpathconf()` pozwala sprawdzić wielkość bufora dla łączy komunikacyjnych, jej wartość jest przechowywana w parametrze `PIPE_BUF`. W przypadku gdy zapisujemy liczbę bajtów mniejszą niż `PIPE_BUF`, dane zostaną zapisane w kolejności, jeżeli większą, takiej gwarancji nie ma, i dane nieobowiązkowo zostaną zapisane do potoku po kolei.

### 2. Praca z łączyami komunikacyjnymi

1. `dup()` / `dup2()` – tworzą kopię deskryptora pliku `oldfd`. Stary i nowy deskryptor mogą być używane zamiennie. Współdzielą one blokady, pozycję pliku i znaczniki; nie współdzielą jednak znacznika `close-on-exec`.  
`dup` używa dla nowego deskryptora nieużywanego deskryptora o najniższym numerze.  
`dup2` powoduje, że `newfd` staje się kopią `oldfd`, zamykając najpierw `newfd`, jeśli jest to potrzebne.

funkcja `close(0)` – zamyka wejście `STDIN`. Jest to potrzebne dla następnej funkcji `dup(fd[0])`, która duplikuje deskryptor wejścia do pierwszego zamkniętego deskryptora, tak więc `stdin` stało tym z potoka.