

W mojej implementacji do komunikacji między procesami wykorzystuję głównie kolejki komunikatów oraz pojedynczego łącza nienazwanego (pipe). Komunikacja ta w kolejności wysłania zapytania wygląda następująco:

- Podproces testera wczytuje zapytanie i wysyła je w kolejce do podprocesu validatora
- Podproces validatora odbiera zapytanie i wysyła je w kolejce do głównego procesu run
- Następnie główny proces run odbiera zapytanie, wykonuje fork() zlecając dla swojego potomka ocenę zapytania i wysłanie wyniku do validatora samemu odbierając kolejne zapytania
- Potomek procesu run wysyła wynik do validatora, który aktualizuje statystyki i odsyła odpowiedź na zapytanie do głównego procesu testera, który wysłał zapytanie.

W przypadku wystąpienia słowa "!" procesy przekazują je sobie w identycznej kolejności jak wyżej, oraz następują dodatkowe dwie komunikacje:

- Podproces validators wysyła liczbę odebranych komunikatów do głównego procesu za pomocą łącza nienazwanego
- Jeśli to główny proces tester odebrał "!" od validatora, to wysyła sygnał do swojego syna, który kończy wczytywanie i przesyła liczbę wysłanych zapytań
- Jeśli to podproces tester napotkał "!" podczas wczytywania, to traktuje to jako normalne zapytanie dodatkowo wysyłając do swojego ojca liczbę wysłanych zapytań

Wszystkie wykorzystywane przeze mnie kolejki są jednostronne, to znaczy proces może tylko czytać lub tylko pisać do danej kolejki. Pozwala mi to na dość proste określenie odpowiedzialności za kolejki: proces czytający z danej kolejki ma obowiązek ją usunąć po skończeniu pracy z nią. Tą konwencję zachowałem we wszystkich plikach.

Poza kolejkami i jednym łączem nienazwanym korzystam też z własnej implementacji (bardzo uproszczonej) tablicy dynamicznej, wykorzystuję również niewielkie (do 50000 elementów) tablice statyczne.

Do zapewnienia kontroli błędów wykorzystałem bibliotekę "err.h" przedstawioną nam na laboratorium.