Laboratoria 6

Krzysztof Żywiecki

24 kwietnia 2020

Ćwiczenie 1

W ćwiczeniu pokazany jest przykład zastosowania funkcji pipe do ustalenia komunikacji między procesami. W przykładowym programie następuje zapis, a następnie odczyt danych z pośrednika. Po użyciu funkcji fork trzeba zamknąć w procesach odpowiednie deskryptory. Strona zapisująca oszczędza w ten sposób alokacje. Strona odczytująca z kolei musi zamknąć swój pipe do zapisu, gdyż przeciwna strona musi po skończeniu zapisu zasygnalizować EOF, co nie jest możliwe jak istnieje kilka zapisujących stron.

Funkcja *fpathconf* dostarcza nam wielu informacji na temat ograniczeń systemu. Wielkość bufora określa ile bajtów może zostać wpisanych do *pipe*

Ćwiczenie 2

Program demonstracyjny w bardzo sprytny sposób łączy *pipe[0]* z wejściem standardowym do procesu. Funkcja *dup* tworzy kopię deskryptora na najniższy możliwy indeks. Po wykonaniu *close(0)* jest to właśnie standardowe wejście. W programie dodano zamykanie deskryptorów: 0 w wątku rodzica, oraz 1 w wątku potomnym jak w poprzednim zadaniu. Dodatkowo po wykonaniu *dup* można zwolnić w wątku potomnym deskryptor 0.

Ćwiczenie 3

//Dokończone w piątek

W zadaniu mamy do wykonania 3 programy.

Pierwszy wymaga do stworzenia dwóch procesów potomnych wraz z metodami komunikacji między nimi. Proces główny wczytuje całość pliku i przekazuje go do potomnych procesów. Te odczytują tekst linijka po linijce, i liczą odpowiednio linijki.

Drugi program tworzy dwa procesy potomne. Pierwszy z procesów wykonuje proces seq z odpowiednimi argumentami. Drugi z nich odczytuje kolejne liczby i wysyła je przemnożone przez 5 do procesu głównego .

Trzeci program wykonuje po kolei 4 procesy: wczytujący nazwy użytkowników, obcinający je do nazwy użytkownika, sortujący je alfabetycznie i biorący tylko unikatowe wartości.