**Przedmiot:** Systemy operacyjne - laboratorium

**Tematyka:** Komunikacja międzyprocesowa (ang. *Inter-process communication*)

**Proszę wybrać jedno zadanie z trzech poniższych**

**Zadanie 1. chat (cc, sh)**

Dobierz odpowiednią technikę komunikacji miedzyprocesowej, zaprojektuj i zaimplementuj program, który realizuje funkcjonalność ‘chat’ w wersji:

* Rozwiązanie podstawowe: na jednym komputerze (dwie osoby zalogowane w sensie ssh/telnet mogące ze soba wymieniać wiadomości) - max 4.0,
* Rozwiązanie rozszerzone: więcej niż dwie osoby - max 5.0.
* Rozwiązanie (\*) na dwóch komputerach połączonych przez sieć Ethernet - max 4.5 (więcej niż 2 osoby: max 5.0).

**Zadanie 2. guess-my-word-2 (można wykorzystać algorytm z guess-my-word lab 2.)**

Dobierz odpowiednią technikę komunikacji miedzyprocesowej, zaprojektuj i zaimplementuj program, który realizuje funkcjonalność gry ‘zgadnij słowo’ (‘wisielec’) w wersji:

* Rozwiązanie podstawowe: na jednym komputerze (dwie osoby zalogowane w sensie ssh/telnet mogące ze soba wymieniać wiadomości) - max 4.0,
* Rozwiązanie rozszerzone: więcej niż jedna osoba - max 5.0.
* Rozwiązanie (\*) na dwóch komputerach połączonych przez sieć Ethernet - max 4.5 (więcej niż 2 osoby: max 5.0).

Słownik do gry można pobrać z: <https://sjp.pl/sl/po.phtml>

Oznaczenia liter, przykładowo:

* 1 - a, i, o, e, m, n, s, itp.
* 2 - p, y, , itp.
* 3 - t, k, l, itp.

Stosować tylko małe litery.

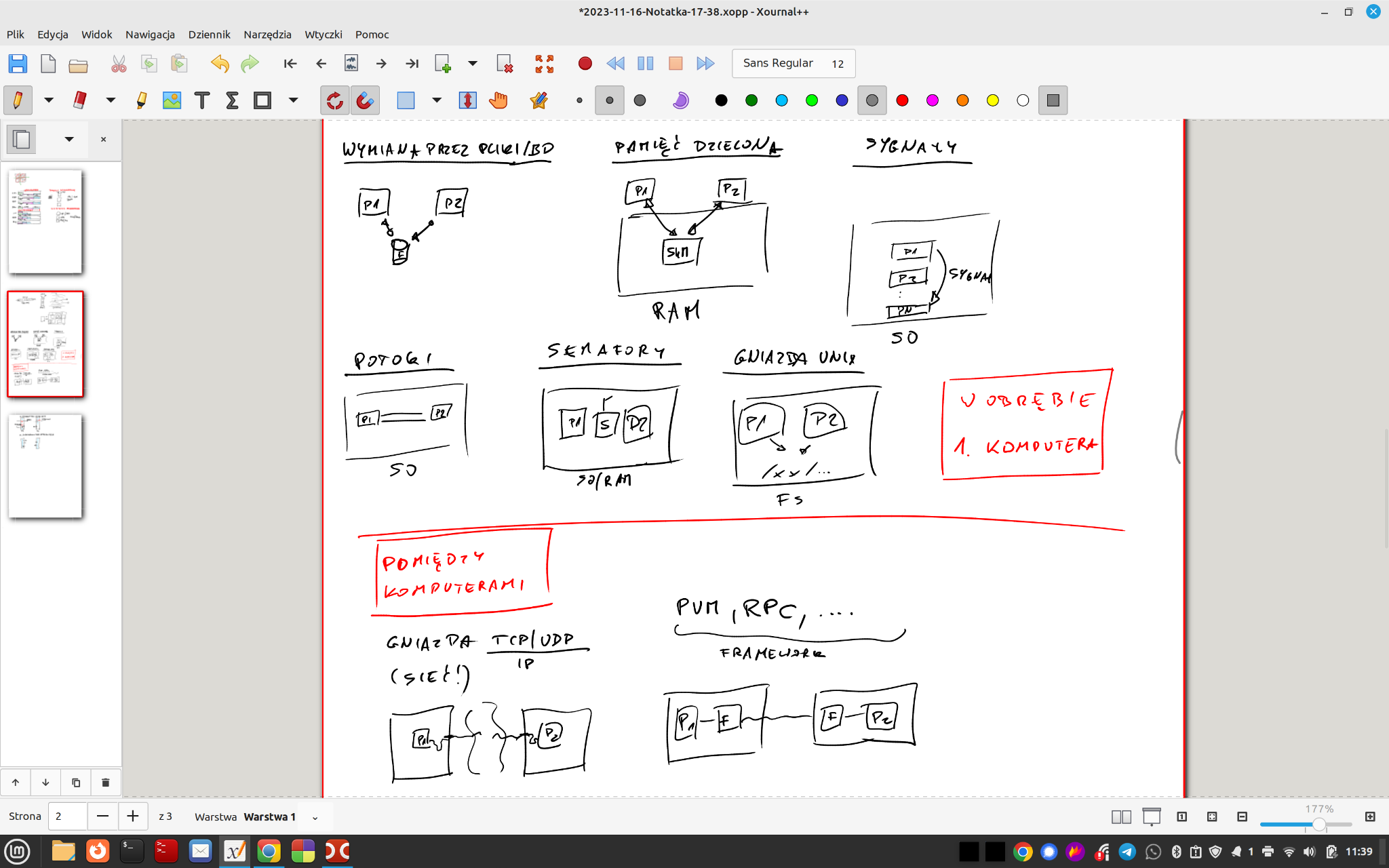
Słowo losuje komputer. W przypadku większej liczby graczy należy zapewnić odpowiednią synchronizację.

**Zadanie 3. papier-nożyce-kamień**

Analogicznie do zadania skryptowego z laboratorium 1 (zadanie 6), dla 2 lub więcej użytkowników ustalając własny sposób punktacji.

**Techniki do wyboru:**

* Wymiana przez pliki
* Pamięć dzielona (było na wykładzie)
* Sygnały
* Potoki nazwane lub nienazwane
* Semafory
* Kolejki
* Gniazda dziedziny UNIX
* Gniazda udp/tcp
* RPC



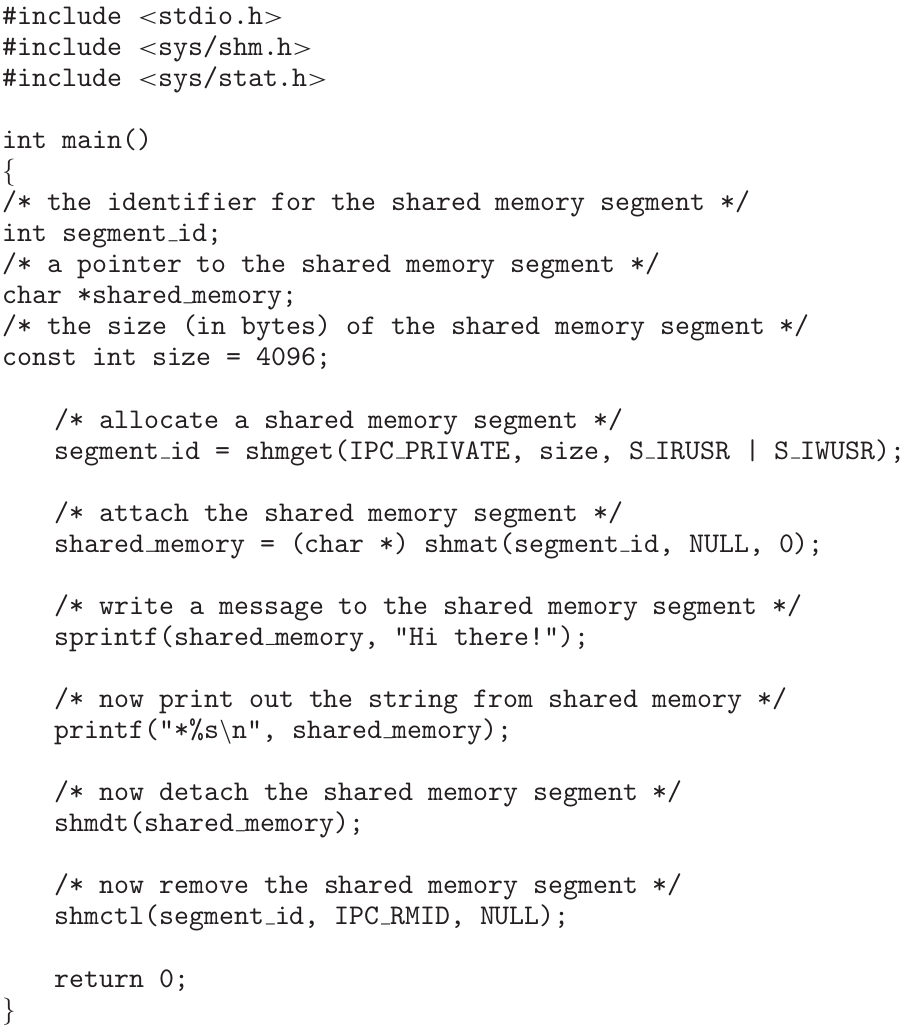
**Uwaga!**

Wymienione są różne możliwości komunikacji miedzyprocesowej, ale nie wszystkie się nadają do rozwiązania tego problemu.

**Literatura:**

* <http://komputeks.pl/unix-programowanie-uslug-sieciowych-tom-komunikacja-miedzyprocesowa-p-595.html>
* W katalogu z instrukcjami laboratoryjnymi są dwa pliki:
  + Instrukcja PUS-01-Gniazda… .pdf
  + Programy przykładowe.

Przykład programu do komunikacji przy pomocy pamięci dzielonej:



S\_IRGRP

Permits read access when the effective group ID of the caller matches either shm\_perm.cgid or shm\_perm.gid.

S\_IROTH

Permits other read access.

S\_IRUSR

Permits read access when the effective user ID of the caller matches either shm\_perm.cuid or shm\_perm.uid.

S\_IWGRP

Permits write access when the effective group ID of the caller matches either shm\_perm.cgid or shm\_perm.gid.

S\_IWOTH

Permits other write access.

S\_IWUSR

Permits write access when the effective user ID of the caller matches either shm\_perm.cuid or shm\_perm.uid.

W funkcji shmget mamy klucz: IPC\_PRIVATE oznaczający, że proces sam dla siebie wykorzystuje pamięć dzieloną. Aby dwa procesy mogły się komunikować, należy zastąpić go liczbą (na przykład: 12345).

Podpowiedź do wykorzystania potoków: wykład 5, slajd 19.