

Systemy operacyjne 2		
Informatyka S1	Semestr 4	2019/2020
Laboratorium 6		

Wstęp teoretyczny:

- Obsługa gniazd (f-cje **socket**, **connect**, **bind**, **listen**, **accept**, **read**, **recv**, **write**).
- Manipulowanie adresami i portami (f-cje **gethostbyname**, **getaddrinfo**, **htons**).
- Obsługa wielu połączeń z wykorzystaniem multipleksingu (**poll**)
- Tworzenie demonów.

Zadanie:

- Napisać dwa programy, które będą się komunikować za pomocą gniazd internetowych. Jeden będzie pełnił rolę serwera, drugi będzie aplikacją kliencką.
- **Serwer** jest demonem, który po uruchomieniu ma odciąć się od terminala sterującego. Program serwera uruchamiany jest obowiązkowo z jedną z dwóch opcji:
 - **-p port** która spowoduje, że serwer zacznie działać w tle nasłuchując na połączenia klientów na porcie **port** na wszystkich dostępnych interfejsach; w systemie może działać jednocześnie tylko jedna instancja naszego serwera, próba ponownego jego uruchomienia powinna być sygnalizowana odpowiednim komunikatem;
 - **-q** która spowoduje, że zostanie sprawdzone czy w systemie działa aktualnie instancja serwera i, jeżeli okaże się to prawdą, zostanie ona zamknięta.
- **Klient** (program użytkownika) uruchamiany jest z obowiązkowymi czterema parametrami **-a adres_ip** , **-p port** , **-s string** oraz **-o operation**, wskazującymi do którego serwera chcemy się podłączyć oraz co serwer ma dla nas zrobić. Po podłączeniu do serwera klient przesyła do niego łańcuch znaków **string** oraz operację do wykonania **operation**. Dozwolone operacje to **tolower**, **toupper** i **invert**.
- Serwer odpowiada klientowi zgodnie z otrzymanym żądaniem przysyłając z powrotem zmodyfikowany łańcuchem znaków (wszystkie litery zostały pomniejszone, wszystkie litery zostały powiększone lub litery zostały zapisane w odwrotnej kolejności).
- Po otrzymaniu odpowiedzi klient wyświetla ją i kończy swoje działanie.
- Należy uwzględnić możliwość wystąpienia sytuacji błędnych i odpowiednio na nie reagować (np. próba wysłania komunikatu do klienta, który się zakończył).
- W serwerze wykorzystać mechanizm multipleksingu.

Uwaga! Kody źródłowe programów (2 pliki) po oddaniu prowadzącemu zajęcia laboratoryjne muszą zostać przesłane na adres so2@zut.edu.pl:

- pliki z kodami źródłowymi muszą mieć nazwy **lab06.server.c** i **lab06.client.c**
- do maila musi być dołączony plik **manifest.json**. Plik manifestu dla konkretnego zadania można pobrać ze strony z materiałami. W pliku manifestu należy wpisać poprawne wartości **email**, **imię**,

nazwisko, indeks i grupa. W polach **kompilacja** i **uruchomienie** wpisujemy polecenia, za pomocą których dokonujemy kompilacji i testowego uruchomienia. W wszystkich polach są wprowadzone przykładowe dane,

- mail musi zostać wysłane z poczty uczelnianej (**zut.edu.pl**),
- nagłówek maila powinien zawierać przynajmniej informację o numerze zadania, np.: **SO2 S1 LAB06**

Dostarczone kody źródłowe będą analizowane pod kątem występowania plagiatów, kompilowane i uruchamiane (zgodnie z informacją przesłaną w manifeście). Niewysłanie wiadomości, wysłanie jej w formie niezgodnej z powyższymi wymaganiami lub wysłanie kodu, który się nie kompiluje i nie uruchamia będzie traktowane jako brak programu i skutkowało otrzymaniem za niego oceny niedostatecznej.