

Systemy operacyjne

Lista zadań nr 10

Na zajęcia 17 grudnia 2019 i 8 stycznia 2020

Należy przygotować się do zajęć czytając następujące rozdziały książek lub publikacji:

- Computer Systems: A Programmer's Perspective (wydanie trzecie): 11.4
- Unix Network Programming: The Sockets Networking API (wydanie trzecie): 4 i 8

Zadanie 1. Na podstawie §4.1 i §8.1 książki „Unix Network Programming: The Sockets Networking API” przedstaw diagram komunikacji klient-serwer za pomocą protokołu TCP i UDP z użyciem, odpowiednio, interfejsu **gniazd strumieniowych** i **datagramowych**. Która ze stron komunikacji używa **portów ulotnych** (ang. *ephemeral*)? Czemu w przypadku protokołu UDP do komunikacji należy używać wywołań systemowych **recvfrom(2)** i **sendto(2)** (o ile wcześniej nie wykonano **connect(2)**) zamiast **read(2)** i **write(2)**?

Ściągnij ze strony przedmiotu archiwum «so19_lista_10.tar.gz», a następnie zapoznaj się z jego zawartością. Na drugiej godzinie zajęć studenci będą mieli szansę zaprezentować wybrane fragmenty własnej implementacji projektu She11. Za taką aktywność prowadzący może przedzielić do 3 punktów bonusowych.

Zadanie 2 (P). Zmodyfikuj program «hostinfo.c» w taki sposób, aby wyświetlał zarówno adresy IPv4, jak i IPv6 dla danej nazwy serwera. Dodatkowo należy przekształcić nazwę usługi przekazanej jako opcjonalny trzeci parametr programu na numer portu. Poniżej przykład:

```
# hostinfo www.google.com https
216.58.215.68:443
[2a00:1450:401b:803::2004]:443
```

Co należałoby zrobić, żeby program rozpoznawał usługę o nazwie «tftp»?

Zadanie 3 (P). Zapoznaj się z kodem źródłowym serwera «echoserver.c» i klienta «echoclient.c» usługi podobnej do «echo». Twoim zadaniem jest taka modyfikacja serwera, by po odebraniu sygnału «SIGINT» wydrukował liczbę bajtów odebranych od wszystkich klientów, po czym zakończył swe działanie.

Używając programu «watch» uruchom polecenie «netstat -ptn», aby obserwować stan połączeń sieciowych. Wystartuj po jednym procesie serwera i klienta. Wskaż na wydruku końce połączenia należące do serwera i klienta. Następnie wystartuj drugą instancję klienta. Czemu nie zachowuje się ona tak samo jak pierwsza? Co zmieniło się na wydruku z «netstat»?

Zadanie 4 (P). Serwer z poprzedniego zadania nie radził sobie ze współbieżną obsługą wielu połączeń. Serwer z pliku «echoclient-fork.c» naprawia to poważne ograniczenie z użyciem wywołania «fork». Zadaniem głównego procesu jest odbieranie połączeń i delegowanie ich obsługi do podprocesów.

Proces serwera musi zliczać liczbę bajtów odebranych od klientów. W tym celu przydziela dzieloną pamięć anonimową, w której przechowuje tablicę «client». Przy starcie podprocesu umieszcza w tablicy odpowiedni wpis za pomocą procedury «addclient». Żeby uniknąć wyścigów każdy podproces zwiększa własny licznik «nread». Po zakończeniu podprocesu należy wywołać procedurę «delclient», która doda zawartość prywatnego licznika klienta, do globalnego licznika serwera.

W dowolnym momencie działanie serwera może zostać przerwane przy pomocy sygnału «SIGINT». Należy wtedy poczekać na zakończenie podprocesów i wydrukować zawartość globalnego licznika serwera. Poniżej przykładowy wydruk z sesji serwera:

```
# ./echoserver-fork 8000
[9047] Connected to localhost:36846
[9105] Connected to localhost:36850
[9047] Disconnected!
^C
Server received quit request!
[9105] Disconnected!
Server received 22 bytes
#
```