

Programowanie sieciowe

Tematy projektów

Grupa projektowa może maksymalnie liczyć trzy osoby. Temat projektu jest ogólnym zarysem tematu. Grupa projektowa powinna określić szczegółową funkcjonalność programu (projekt programu) a następnie przygotować opis i zatwierdzić u prowadzącego (czas do końca kwietnia). Programy powinny wykorzystywać adresację multicast (zwykle do wyszukania usługi) i unicast. W programie należy uwzględnić przesyłanie danych w formacie binarnym, organizując przesyłane dane np. za pomocą formatu TLV. Serwer powinien działać w trybie „demona” z logowaniem do plików systemowych. Należy użyć funkcji do przekształcania adresów na nazwy.

1. Implementacja protokołu: Synchronizing Internet Clock frequency protocol (sic) (draft-alavarez-hamelin-tictoc-sic-02) – multicast, UDP, SCTP
2. Napisać program typu klient-serwer (serwer współbieżny), który będzie służył do wymiany plików (cos w rodzaju FTP) i operacji na plikach. Program ma działać w trybie interaktywnym.
3. Napisać program typu klient-serwer (serwer współbieżny), który będzie realizował wykonywanie zdalnych poleceń powłoki shell w trybie interaktywnym (nie wsadowym). Po stronie serwera użyć funkcji: pipe, exec, fork i uruchamiania poleceń i przekazywania wyników do procesu serwera. (obsługa protokołów TCP i SCTP)
4. Gra sieciowa: statki. Serwer do kojarzenia klientów. Klienci grają bezpośrednio lub z pośrednictwem serwera.
5. Gra sieciowa: szachy. Serwer do kojarzenia klientów. Klienci grają bezpośrednio lub z pośrednictwem serwera.
6. Gra sieciowa: kółko i krzyżyk. Serwer do kojarzenia klientów. Klienci grają bezpośrednio lub z pośrednictwem serwera.
7. Napisać program typu komunikator tekstowy używając protokołu TCP, UDP i SCTP.
8. Implementacja uproszczonego protokołu typu MQTT – SCTP, TCP, UDP
9. Opracować ćwiczenie do testowania opcji gniazd **IP_RECVERR / IPV6_RECVERR**
10. Opracować ćwiczenie do testowania odbierania i ustawiania wartości pól nagłówka protokołu IP używając funkcji `recvmsg()` i `sendmsg()`
11. Zaimplementować z użyciem protokołu UDP uproszczoną wersję programu `tracpath/traceroute` – biblioteka
12. Opracować program typu klient-serwer UDP, które będzie pokazywał szacowanie RTT zgodnie z publikacją "V. Jacobson, „Congestion Avoidance and Control”, Originally Published in: Proc. SIGCOMM '88, Vo118 No. 4, August 1988". W serwerze zaimplementować wprowadzanie opóźnień i gubienie pakietów. Opracowanie ćwiczenia laboratoryjnego.
13. Napisać program/bibliotekę typu IPERF dla protokołu UDP, SCPT, TCP z obsługą adresów multicast (dynamiczny port serwera) i raportowaniem wyników po stronie klienta.
14. Program do testowania jakości połączenia w sieci TCP/IP z wykorzystaniem protokołów OWAMP i TWAMP
15. Analizator ruchu TCP/UDP/SCTP z wykorzystaniem biblioteki `libpcap`

16. Program do ataku typu DoS/DDoS protokołów transportowych TCP i SCTP.
17. Program do wykrywania działających usług (otwartych portów) na serwerze – diagnostyka sieci TCP/IP
18. Własny pomysł – do konsultacji z prowadzącym