

# Technologie Sieci Komputerowych - laboratorium 2006/2007 studia dzienne

## 1. Treść zadania:

Stworzyć oprogramowanie jednocześnie transferujące plik z jednego hosta (serwera) do wielu (klienci). Hosty nie muszą znajdować się w tej samej podsieci IP. Należy wykorzystać multicastowe rozgłaszanie pakietów.

## 2. Wymagania i ocena

### – ocena max: 60%:

1. Wszelkie konieczne do transmisji adresy i numery portów są znane, pozostałe wartości przekazywane do programu jako argumenty w wierszu polecenia
2. Dane przesyłane są w sposób pewny
3. Oprogramowanie reaguje na „wypadnięcie” klienta
4. Wielkość przesyłanych bloków jest znana, ustalana podczas uruchamiania serwera i klientów.
5. Transmitowany plik odczytywany jest z dysku lokalnego przez serwer i zapisywany lokalnie przez klientów.

### – ocena max: 70%

1. Plik konfiguracyjny (tekstowy): XML lub opcja=wartość;
2. TCP-Proxy-Multicast: Serwer dodatkowo może pełnić rolę tzw. proxy. Plik nie jest odczytywany z jego dysku lokalnego lecz otrzymywany przez sieć od jednego z hostów (Transmitter) z wykorzystaniem protokołu TCP.

### – ocena max: 80%

1. Automatyczne rozpoznawanie adresów/portów serwera i parametrów transmisji (np. wielkość przesyłanych bloków)
2. Optymalizacja protokołu pod kątem wydajności i niezawodności

### – ocena max: 90%

1. Wprowadzenie wielowątkowości
2. Kompresja przesyłanych danych

### – ocena max: 100%

1. Uwierzytelnianie

## 3. Platformy:

1. Język programowania C++
2. System operacyjny Linux
  - a. Fedora Core 6, gcc, wymagany skrypt powłoki wykonujący na żądanie kompilację statyczną lub zwykłą (na podstawie parametru wiersza polecenia lub wyboru interaktywnego)
3. System operacyjny Windows

- a. Oprogramowanie musi zostać skompilowane w DevC++

#### **4. Dokumentacja**

Sprawozdanie w formie elektronicznej musi składać się z:

1. Strony tytułowej – numer zadania, data, nazwa grupy, skład grupy (numery indeksów).
2. Teorii – opis działania i sposobu wykorzystania w programie funkcji i struktur socketowych (UWAGA! Nie wolno kopiować kompletnej informacji z dokumentacji API. Należy raczej opisać w jakim celu funkcję użyto, jakie przesłano parametry i co one oznaczają oraz jaki był efekt wywołania funkcji.).
3. Szczegółowego projektu klienta, serwera, transmitera - najlepiej w postaci diagramów UML. Opis struktur danych, funkcji, algorytmy, przepływy danych. Uwzględnić model protokołu komunikacyjnego.
4. Specyfikacji protokołów
5. Testów wydajności