

Telekommunikációs Hálózatok

6. gyakorlat

Házi feladat

netcopy alkalmazás

Készíts egy netcopy kliens/szerver alkalmazást, mely egy fájl átvitelét és az átvitt adat ellenőrzését teszi lehetővé CRC vagy MD5 ellenőrzőösszeg segítségével! A feladat során három komponenst/programot kell elkészíteni:

1. Checksum szerver: (fájl azonosító, checksum hossz, checksum, lejárati idő (mp-ben)) négyesek tárolását és lekérdezését teszi lehetővé. A protokoll részletei a következő oldalon.
2. Netcopy kliens: egy parancssori argumentumban megadott fájlt átküld a szervernek. Az átvitel során/végén kiszámol egy md5 checksumot a fájlra, majd ezt feltölti fájl azonosítóval együtt a Checksum szerverre. A lejárati idő 60 mp. A fájl azonosító egy egész szám, amit szintén parancssori argumentumban kell megadni.
3. Netcopy szerver: Vár, hogy egy kliens csatlakozzon. Csatlakozás után fogadja az átvitt bájtokat és azokat elhelyezi a parancssori argumentumban megadott fájlba. A végén lekéri a Checksum szervertől a fájl azonosítóhoz tartozó md5 checksumot és ellenőrzi az átvitt fájl helyességét, melynek eredményét stdoutputra is kiírja. A fájl azonosító itt is parancssori argumentum kell legyen.

Beadás: **TMS-en keresztül, DE még nincs meghirdetve**

Checksum szerver - TCP

- Beszúr üzenet
 - Formátum: szöveges
 - Felépítése: BE|<fájl azon.>|<érvényesség másodpercben>|<checksum hossza bájtyszámban>|<checksum bájtjai>
 - A „|” delimiter karakter
 - Példa: BE|1237671|60|12|abcdefabcdef
 - Ez esetben: a fájlazon: 1237671, 60mp az érvényességi idő, 12 bájt a checksum, abcdefabcdef maga a checksum
 - Válasz üzenet: OK
- Kivesz üzenet
 - Formátum: szöveges
 - Felépítése: KI|<fájl azon.>
 - A „|” delimiter karakter
 - Példa: KI|1237671
 - Azaz kérjük az 1237671 fájl azonosítóhoz tartozó checksum-ot
 - Válasz üzenet: <checksum hossza bájtyszámban>|<checksum bájtjai>
Péda: 12|abcdefabcdef
 - Ha nincs checksum, akkor ezt küldi: 0|
- Futtatás
 - ./checksum_srv.py <ip> <port>
 - <ip> - pl. localhost a szerver címe bindolásnál
 - <port> - ezen a porton lesz elérhető
 - A szerver végtelen ciklusban fut és egyszerre több klienst is ki tud szolgálni. A kommunikáció TCP, csak a fenti üzeneteket kezeli.
 - Lejárat utáni checksumok törlődnek.

Netcopy kliens – TCP alapú

- Működés:
 - Csatlakozik a szerverhez, aminek a címét portját parancssori argumentumban kapja meg.
 - Fájl bájtjainak sorfolytonos átvitele a szervernek.
 - A Checksum szerverrel az ott leírt módon kommunikál.
 - A fájl átvitele és a checksum elhelyezése után bontja a kapcsolatot és terminál.
- Futtatás:
 - `./netcopy_cli.py <srv_ip> <srv_port> <chsum_srv_ip> <chsum_srv_port> <fájl azon> <fájlnév elérési úttal>`
 - <fájl azon>: egész szám
 - <srv_ip> <srv_port>: a netcopy szerver elérhetősége
 - <chsum_srv_ip> <chsum_srv_port>: a Checksum szerver elérhetősége

Netcopy szerver – TCP alapú

- Működés:
 - Bindolja a socketet a parancssori argumentumban megadott címre.
 - Vár egy kliensre.
 - Ha acceptálta, akkor fogadja a fájl bájtjait sorfolytonosan és kiírja a parancssori argumentumban megadott fájlba.
 - Fájlvége jel olvasása esetén lezárja a kapcsolatot és utána ellenőrzi a fájlt a Checksum szerverrel.
 - A Checksum szerverrel az ott leírt módon kommunikál.
 - Hiba esetén a stdout-ra ki kell írni: CSUM CORRUPTED
 - Helyes átvitel esetén az stdout-ra ki kell írni: CSUM OK
 - Fájl fogadása és ellenőrzése után terminál a program.
- Futtatás:
 - `./netcopy_srv.py <srv_ip> <srv_port> <chsum_srv_ip> <chsum_srv_port> <fájl azon> <fájlnév elérési úttal>`
 - <fájl azon>: egész szám ua. mint a kliensnél – ez alapján kéri le a szervertől a checksumot
 - <srv_ip> <srv_port>: a netcopy szerver elérhetősége – bindolásnál kell
 - <chsum_srv_ip> <chsum_srv_port>: a Checksum szerver elérhetősége
 - <fájlnév> : ide írja a kapott bájtokat

CRC, MD5 pythonban

- CRC

```
import binascii, zlib

test_string = "Fekete retek rettenetes".encode('utf-8')

print(hex(binascii.crc32(bytearray(test_string))))
print(hex(zlib.crc32(test_string)))
```

- MD5

```
import hashlib

test_string = "Fekete retek rettenetes".encode('utf-8')

m = hashlib.md5()
m.update(test_string)
print(m.hexdigest())
```

Fájl átvitel

- fájl bináris megnyitása

```
with open („input.txt”, „rb”) as f:  
    ...
```

- read(x) – x bytes

```
...  
f.read(128)  #128 byte-ot fog beolvasni
```

„When size is omitted or negative, the entire contents of the file will be read and returned; it's your problem if the file is twice as large as your machine's memory. „ - python.org

PYTHON SOCKET - PROXY

Feladat

Készítsünk egy egyszerű TCP alapú proxyt (átjátszó). A proxy a kliensek felé szerverként látszik, azaz a kliensek csatlakozhatnak hozzá. A proxy a csatlakozás után kapcsolatot nyit egy szerver felé (parancssori argumentum), majd minden a kienstől jövő kérést továbbítja a szerver felé és a szervertől jövő válaszokat pedig a kliens felé.

Pl: netProxy.py ggombos.web.elte.hu 80

Web browserbe írjuk be: localhost:10000

Feladat – Tiltsunk le valamilyen tartalmat

A SzamHalo-t tartalmazó URL-ek ne legyenek elérhetőek a proxyn keresztül.

A válasz legyen valamilyen egyszerű HTML üzenet, ami jelzi a blokkolást.

megj: header: „HTTP/1.1 404 Not Found\n\n”

Proxy

- Készítsünk a számítógéphez egy proxy-t, ami értelmezi a kienstől kapott TCP kéréseket, az első operandust megszorozza kettővel, a másodikhoz pedig hozzáad egyet. A módosított kérést elküldi a TCP servernek, majd az eredményt visszaküldi a kliensnek.

Feladat 3 – fájlátvitel UDP felett

Fájlátvitel megvalósítása úgy, hogy a fájl letöltése UDP felett legyen megoldva. Készüljünk fel arra, hogy az átvitel során csomagvesztés, vagy sorrend csere is történhet! Az UDP szerver portját szabadon definiálhatjuk!

A hibakezeléshez egy javaslat:

Max. 1000 bájtanként UDP csomagokban elkezdjük átküldeni a fájl tartalmát. Minden csomag egy pár bájtos fejléccel indul, amiben jelezzük, hogy az utolsó darab-e, amit átküldtünk, továbbá egy másik mező jelzi a byteoffset-et a fájl elejétől. Működés:

- Ha a kliens kapott egy adatcsomagot, akkor egy nyugtacsomagot küld vissza.
- A nyugtacsomag fogadása után a szerver, küldi a következő adatcsomagot.
- Ha nem jön nyugta, akkor T idő után újraküldi a korábbi adatcsomagot. (pl. $T=200\text{ms}$)
- Ha nyugta veszik el, akkor a vevő az offset alapján el tudja dönteni, hogy egy új adatcsomag, vagy egy korábbi duplikátuma érkezett-e.
- Ha az utolsó csomag is megérkezett, akkor a kliens nyugtázza azt is és lezárja a fájlba írást. A szerver az utolsó nyugta után befejezi az átvitelt.

MapBank zh gyakorlás

- 3 script: naviClient,naviServer,mapBank
- A naviClient 4 számot kap a parancssorról (2 koordináta, óra, perc), ezeket + hard codeolva a Neptun kódodat becsomagolja és UDP-n keresztül elküldi a naviServernek. A naviServer kiírja a kapott infokat a konzolra majd egy TCP kapcsolaton keresztül továbbítja a mapBank-nak.
- A mapBank megnézi, hogy megvan-e neki eltárolva, hogy mennyi idő alatt lehet eljutni a megadott koordinátákra (a kiindulási koordinátáktól most eltekintünk) a megadott időpontban indulva.
- Amennyiben nem, random generál egy értéket amit elküld válaszként, ha igen, akkor a tárolt választ küldi. A naviServer kiírja a konzolra a kapott választ is és továbbítja azt a naviClientnek.

Netmask

- Alhálózat címeinek leírása.

Address (Host or Network)	Netmask (i.e. 24)	Netmask for sub/supernet (optional)
<input type="text" value="192.168.0.1"/>	<input type="text" value="16"/>	move to: <input type="text"/>
<input type="button" value="Calculate"/>	<input type="button" value="Help"/>	

Address:	192.168.0.1	11000000.10101000 .00000000.00000001
Netmask:	255.255.0.0 = 16	11111111.11111111 .00000000.00000000
Wildcard:	0.0.255.255	00000000.00000000 .11111111.11111111
=>		
Network:	192.168.0.0/16	11000000.10101000 .00000000.00000000 (Class C)
Broadcast:	192.168.255.255	11000000.10101000 .11111111.11111111
HostMin:	192.168.0.1	11000000.10101000 .00000000.00000001
HostMax:	192.168.255.254	11000000.10101000 .11111111.11111110
Hosts/Net:	65534	(Private Internet)

Feladat 3

- Hány cím elérhető a következő netmaskokkal és adjuk meg a minimális és maximális címet:
 - 188.100.22.12/32
 - 188.100.22.12/20
 - 188.100.22.12/10

<http://jodies.de/ipcalc>

VÉGE
KÖSZÖNÖM A FIGYELMET!