

Praktika: Echo aplikazioa TCP erabiliz

Deskribapena:

Praktika honetan ikasleak Echo bezero eta zerbitzari bat inplementatu beharko ditu *Python* programazio lengoaian. Echo aplikazioa oso aplikazio sinplea da: zerbitzariak bezerotik jasotzen duen informazio berbera bidaltzen dio bueltan bezeroari. Horregatik dauka aplikazioak Echo (oihartzuna) izena. Praktika honetan garraio mailako zerbitzu bezala TCP erabili beharko da.

Materiala:

Praktika hau egiteko ikasleak hainbat baliabide lagungarri izango ditu:

1. Bezero eta zerbitzariaren iturburu kodearen zatitxo bana (*echo_bez_tcp.py* eta *echo_zer_tcp.py*). Dena den, fitxategiak ia hutsik daude. [Hauek dira ikasleak aldatu beharko dituen fitxategiak.](#)
2. Honetaz aparte Echo zerbitzari bat izango dugu martxan *dif-linuxserver.ehu.es* makinan (158.227.106.30) 50007 portuan. Honela ikasleak urruneko makina batekin frogatu ahal izango du bere bezeroa.

Aplikazio protokoloa

Aplikazio hau bereziki sinplea da, baina hala ere, bezero eta zerbitzariaren arteko komunikazio egokia bermatzeko badira definitu beharreko hainbat kontu.

Aplikazioaren izaera dela eta ez dago komando, erantzun edo parametririk definitu beharrik. Definitu beharreko bakarra nolabaiteko prozedura orokorra da. Prozedura TCP konexio bat ezarriz hasiko da. Puntu honetatik aurrera bezeroak datuak bidali ditzake eta zerbitzariak datuak irakurri ahala inongo aldaketarik gabe bidaliko dizkio bezeroari. Prozedura amaitzeko bezeroak TCP konexioa itxiko du.

Protokoloaren definizio ofiziala [RFC862](#) dokumentuan aurki dezakezu.

Zerbitzaria

Aplikazio honetako zerbitzariak ez dauka berezitasun handirik. TCP konexio batetik jasotzen duen guztia buffer batean gorde eta dagoen moduan bidaliko du konexio beretik. Echo aplikazioak normalean 7 portua erabili arren guk 50007 portua erabiliko dugu, normalean 7 portuan irekiera pasibo bat egiteko administratzaile baimenak behar dira eta.

Bezeroa

Inplementatu beharreko bezeroak sarrera estandarretik irakurriko ditu datuak (*input* funtzioarekin) eta lerroak banan bana bidaliko dizkio zerbitzariari UTF-8 kodeketa erabiliz (erabiltzaileak zerbitzariaren identifikazioa, DNS izena edo IP helbidea, pasako dio argumentu bezala). Bidaltzen duen lerro bakoitzeko erantzuna jasotzeko zain geratuko da eta behin mezua jasotakoan pantailan idatziko du. Kontuz hemen, TCPk ez baitu bermatzen espero diren karaktere guztiak irakurketa bakarrean jasoko direla! Dena den, bezeroak badaki zenbat karaktere jaso behar dituen (bidalitako kopuru bera) eta beraz, behar adina irakurketa egin beharko dira.

Bezeroak prozedura amaitutzat eman eta TCP konexioa itxiko du sarrera estandarretik mezu hutsa irakurtzean (RETURN tekla bakarrik sakatuta) edo erroreren bat gertatzen denean.

Lan egiteko dinamika

1. Bezeroa inplementatzea eta urruneko makinan dagoen zerbitzariarekin probatzea.
2. Zerbitzaria inplementatzea eta inplementatutako bezeroarekin probatzea.

“echo_bez_tcp.py” fitxategia:

```
#!/usr/bin/env python3

import socket, sys

PORT = 50007

# Egiaztatu argumentu bat pasa dela.
if len(sys.argv) != 2:
    print("Erabilera: {} <zerbitzaria>".format(sys.argv[0]))
    exit(1)

"""IKASLEAK BETETZEKO:
Sortu socketa eta konektatu zerbitzariarekin.
"""

print("Sartu bidali nahi duzun mezua (hutsa bukatzeko):")
while True:
    mezua = input()
    if not mezua:
        break
    """IKASLEAK BETETZEKO:
    Bidali irakurritakoa eta erantzuna jaso.
    Idatzi irteera estandarrean jasotakoa.
    ***Gogoratu irakurketa bakarrean erantzun osoa jasoko dugunaren
    bermerik ez dagoela!
    """
    """IKASLEAK BETETZEKO:
    Itxi socketa.
    """
```

“echo_zer_tcp.py” fitxategia:

```
#!/usr/bin/env python3

import socket

PORT = 50007

"""IKASLEAK BETETZEKO:
Sortu socket bat, esleitu helbide bat eta ezarri entzute socket
bezala.
"""

while True:
    """IKASLEAK BETETZEKO:
    Onartu konexio berri bat.
    Bezeroak konexioa itxi arte jaso mezuak eta erantzun datu
    berdinak bidaliz.
    """
    """IKASLEAK BETETZEKO:
    Itxi entzute socketa.
    """
```

Ariketa osagarriak

- (1) Zerbitzarian bezeroaren socketaren IP helbidea eta portu zenbakia eskuratu eta adierazi, `accept()` funtzioak itzultzen duen bigarren emaitza erabili gabe. Aztertu `getpeername()` funtzioa `socket` moduluan.
- (2) Zerbitzarian mezuak jasotzeko bufferraren tamaina (`recv()` funtzioaren parametroa) txikia jarri eta probatu bezerotik bidaltzen mezu luzeak bidaltzen. Aztertu badabilen edo ez, eta saiatu ulertzen zergatia.