

Δίκτυα Υπολογιστών II

Ακολουθεί επεξήγηση των 2 αρχείων πηγαίου κώδικα, `server.c`, και `client.c`.

`server.c`

Η ροή του προγράμματος είναι η ακόλουθη:

1. Δημιουργία μεταβλητής για την πρόσβαση σε ένα socket.
 - a. Οι παράμετροι που δίνονται δείχνουν στην συνάρτηση `socket` τι πρωτόκολλα θα χρησιμοποιηθούν σε κάθε layer (TCP, IPv4).
2. Ύστερα παρέχονται πληροφορίες για τον server.
 - a. Συγκεκριμένα ότι θα είναι διαθέσιμος σε hosts μέσω IPv4, και θα είναι στην πόρτα 8080.
3. Μετά δημιουργεί το socket με την χρήση της `bind` και των στοιχείων που δόθηκαν.
4. Ο server ακούει για requests μέχρι να τερματιστεί το πρόγραμμα με `ctrl+C`
5. Ύστερα για κάθε request που λαμβάνει:
 - a. Δημιουργεί ένα νέο socket με την χρήση της `accept` για να ανταλλαχθεί η πληροφορία.
 - b. Ανοίγει ένα αρχείο με διαφορετικό αριθμό για κάθε σύνδεση, ώστε να γράψει την πληροφορία
 - c. Διαβάζει τα bytes που στέλνει ο client και τα γράφει στο αρχείο, μέχρι να λάβει ένα μη πλήρες πακέτο (που σημαίνει ότι δεν υπάρχουν άλλα bytes να σταλούν με επόμενο πακέτο).
 - d. Στέλνει στον client πόσα bytes λήφθηκαν και κλείνει την σύνδεση.
 - e. Εμφανίζει στην κονσόλα πόσα bytes γράφτηκαν και σε ποιο αρχείο.

Όλα τα βήματα έχουν ελέγχους ότι οι ενέργειες εκτελέστηκαν ορθά, αλλιώς το πρόγραμμα τερματίζεται. Στην περίπτωση που μία σύνδεση δεν γίνει `accept` ορθά, το σφάλμα απλώς αγνοείται και περιμένει άλλες νέες συνδέσεις.

`client.c`

Το `client.c` λειτουργεί με τον ακόλουθο τρόπο:

1. Παρομοίως με τον server, δημιουργεί μεταβλητή για το socket.
2. Εισάγει τα στοιχεία του server (όμως η IP address του πρέπει είναι γνωστή (127.0.0.1, loopback address)).
3. Διαβάζει από τον χρήστη ποιο αρχείο `jpg` να στείλει.
4. Επιχειρεί να κάνει σύνδεση με τον server.
5. Στέλνει το αρχείο σε κομμάτια των 2048 bytes.
 - a. Ελέγχει πρώτα αν το αρχείο είναι κενό, αν ο server λάβει άδειο πακέτο σταματάει να λειτουργεί.
6. Τέλος, περιμένει από τον server απάντηση περί πόσα bytes λήφθηκαν. Εμφανίζει πόσα έστειλε ο client και πόσα έλαβε ο server.

Σημειώσεις

Σημειώνεται ότι οι βιβλιοθήκες socket programming στα Windows έχουν διαφορές από αυτές για UNIX συστήματα, επομένως μερικές εντολές θα διαφέρουν μεταξύ σε όνομα και υλοποίηση. Τα συγκεκριμένα εκτελέσιμα που έχουν παραχθεί θα τρέξουν σε UNIX συστήματα, αλλά δεν μπορούν να γίνουν compile από την πηγαίο κώδικα.

Παράδειγμα

Έχει προσφερθεί ένα αρχείο με όνομα send . jpg, θα χρησιμοποιηθεί στο ακόλουθο παράδειγμα.

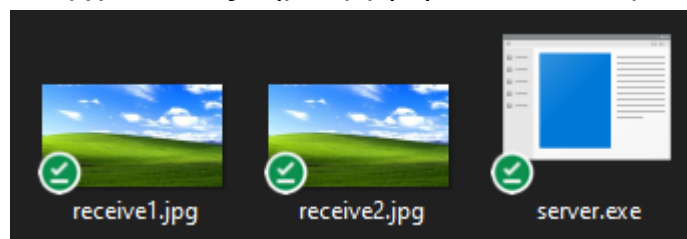
Αρχικά αρχίζω τον server, μετά τρέχω το client και του δίνω το αρχείο να στείλει:

```
Enter image file to be sent:  
C:/send.jpg  
Sent 30589 bytes  
Server received 30589 bytes
```

Αν ξανατρέξω το client και του δώσω το ίδιο αρχείο, ο server θα το αποθηκεύσει με διαφορετικό όνομα:

```
Server started  
Successfully received 30589 bytes, saved to file ./receive1.jpg  
Successfully received 30589 bytes, saved to file ./receive2.jpg
```

Τα αρχεία όντως δημιουργήθηκαν στον ίδιο φάκελο απ'τον οποίο εκτελέστηκε ο server.



Κουλουράς Ιωάννης
Ε20075