

TETPAΛIΖA

Αναφορά Εργασίας Εξαμήνου στο μάθημα Λειτουργικά Συστήματα Ακαδημαϊκού
έτους 2017

Κόκκινος Κυριαζής
Πανεπιστήμιο Δυτικής Μακεδονίας
Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών
AM: 1016
st1016@icte.uowm.gr

Θανάσης Λιατίφης
Πανεπιστήμιο Δυτικής Μακεδονίας
Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών
AM: 1027
st1027@icte.uowm.gr

TETPAΛIZA

21/06/2017

Εισαγωγή

Αντικείμενο της εργασίας είναι η δημιουργία ενός παιχνιδιού τετράλιζας με χρήση του πρωτοκόλλου TCP. Η υλοποίηση έγινε με χρήση της γλώσσας προγραμματισμού C στο πρότυπο Posix σε έκδοση gnu11. Η χρήση του προγράμματος έχει γίνει συγκεκριμένα για Linux-Debian καθώς χρησιμοποιούνται συναρτήσεις και βιβλιοθήκες που βρίσκονται μόνο σε Debian-based Λειτουργικά Συστήματα. Το πρόγραμμα εκτελείτε μόνο μέσω τερματικού καθώς δεν έχει γίνει χρήση βιβλιοθηκών γραφικών όπως OpenGL ή Ncurses. Η εργασία έχει χωριστεί σε δύο μέρη, την εφαρμογή του πελάτη και την εφαρμογή του διακομιστή, τα δύο μέρη θα αναλυθούν παρακάτω. Ο κώδικας της εφαρμογής πελάτη είναι ένα αρχείο C περίπου 500 σειρών, ενώ για την εφαρμογή του διακομιστή είναι ένα αρχείο 500 σειρών. Και οι δύο εφαρμογές μπορούν να λειτουργήσουν σε έναν απλό φορητό υπολογιστή με διπύρηνο επεξεργαστή και 4 GB RAM χωρίς κανένα πρόβλημα ενώ μια τυπική σύνδεση στο διαδίκτυο επαρκεί για την σύνδεση.

Αναλυτική Περιγραφή

Εφαρμογή Server

Το server είναι υπεύθυνο για την αποδοχή παικτών, την εκκίνηση του παιχνιδιού, την επικοινωνία μεταξύ των παικτών, την επεξεργασία των κινήσεων ώστε να βρεθεί ο νικητής, καθώς και την ενημέρωση των παικτών για το τελικό αποτέλεσμα του παιχνιδιού.

Προϋποθέσεις

Το server μπορεί να λειτουργήσει είτε σε τοπικό δίκτυο, είτε μέσω του διαδικτύου, εφόσον το σύστημα είναι κατάλληλα ρυθμισμένο, είτε με συνδιασμό των παραπάνω. Κατά την εκκίνησή του στο τερματικό θα εμφανιστούν οι συσκευές δικτύου που μπορούν να χρησιμοποιηθούν, καθώς και η διεύθυνση IP του υπολογιστή στο δίκτυο.

Είναι επίσης απαραίτητο να δηλώσει ο χρήστης το port στο οποίο θα συνδεθούν τα χρήστες. Αυτό μπορεί να γίνει εισάγοντας το νούμερο της θύρας ως παράμετρο του προγράμματος στο τερματικό. Σε κάθε άλλη περίπτωση, ζητείται κατά την εκκίνηση του server.

Στάδιο εκκίνησης

Αφού δοθούν οι παραπάνω απαραίτητες πληροφορίες, το server πρέπει να διαμορφώσει τη δικτύωση του [1]. Με χρήση των κατάλληλων συναρτήσεων δικτύωσης, δεσμεύει τη θύρα, αρχικοποιεί τις μεταβλητές και τους πίνακες που πρόκειται να χρησιμοποιηθούν και αναμένει συνδέσεις πελατών.

Στη συνέχεια το server χρησιμοποιεί τη συνάρτηση `accept()`, οπότε μπαίνει σε κατάσταση `blocked` έως ότου συνδεθεί ένας παίκτης. Αυτό γίνεται συνολικά δυο φορές. Αφού συνδεθούν δύο παίκτες, δημιουργείται ένα παιχνίδι.

Αναλυτική Περιγραφή

Στάδιο παιχνιδιού

Τόσο η εφαρμογή πελάτη αλλά και διακομιστή, κατά το στάδιο της διεξαγωγής του παιχνιδιού ανταλλάσσουν ορισμένα μηνύματα, τα οποία στέλνονται πριν από κάθε επικοινωνία μεταξύ του πελάτη και διακομιστή. Η διαδικασία αυτή βοηθάει στο συγχρονισμό και την απλούστερη και περισσότερο κατανοητή υλοποίηση. Τα μηνύματα αυτά έχουν μήκος έναν χαρακτήρα και η εφαρμογή του πελάτη αφού τα διαβάσει αναγνωρίζει πως πρέπει να αντιδράσει.

Η λίστα με τα μηνύματα που στέλνονται είναι:

- “1” : Το server θα διαβάσει το όνομα του παίκτη.
- “2” : Το server θα στείλει το όνομα του αντιπάλου.
- “3” : Ενημερώνουμε τον παίκτη πως είναι η σειρά του.
- “4” : Ενημερώνουμε τον παίκτη πως δεν είναι η σειρά του.
- “5” : Ενημερώνουμε τον παίκτη πως έχει νικήσει.
- “6” : Ενημερώνουμε τον παίκτη πως έχει χάσει.
- “7” : Ενημερώνουμε τον παίκτη πως είναι ισοπαλία.
- “8” : Ο παίκτης από τον οποίο διάβασε ο διακομιστή το μήνυμα έχει παραιτηθεί.

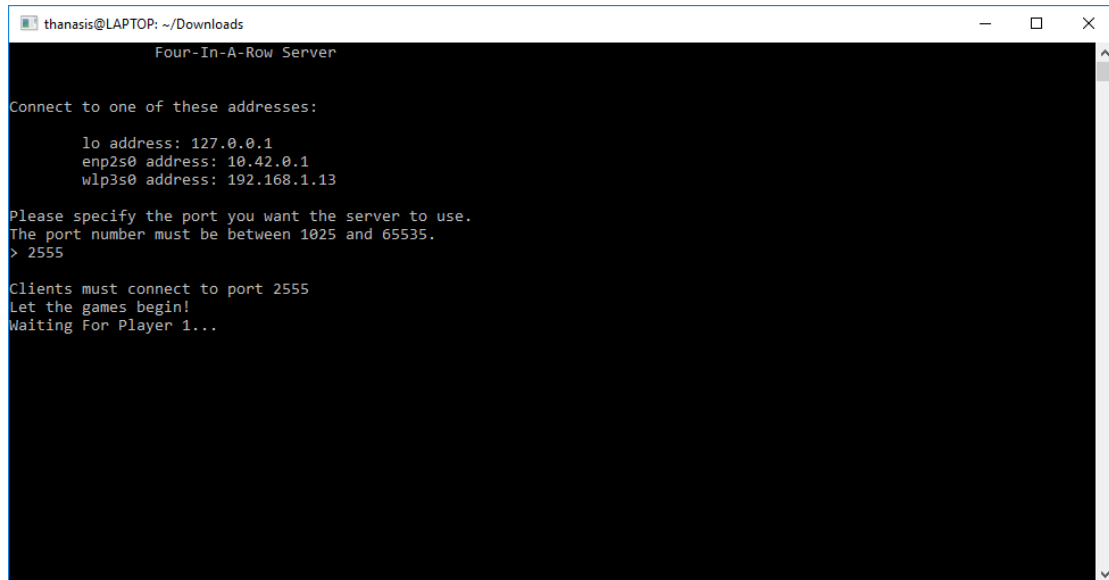
Για παράδειγμα, το server στέλνει στον παίκτη 1 τον χαρακτήρα “3” ώστε να τον ειδοποιήσει ότι είναι η σειρά του να παίξει. Με το που σταλθεί ο χαρακτήρας, το server γνωρίζει ότι θα πρέπει να διαβάσει την κίνηση του παίκτη. Συγχρόνως, το server έχει στείλει και τον χαρακτήρα “4” στον παίκτη δύο. Έτσι, ο παίκτης δύο γνωρίζει ότι δεν είναι η σειρά του και συνεπώς θα περιμένει να διαβάσει από το server την κίνηση που θα κάνει ο αντίπαλος.

Η διαδικασία αυτή επαναλαμβάνεται έως ότου δεν υπάρχουν άλλες πιθανές κινήσεις, δηλαδή το παιχνίδι έχει καταλήξει σε ισοπαλία, ή νικήσει ένας από τους δύο παίκτες.

Αναλυτική Περιγραφή

Στάδιο τερματισμού

Αφού τελειώσει το παιχνίδι, παραμένει το κλείσιμο των ενεργών συνδέσεων, η αποδέσμευση των πόρων και ο τερματισμός του server.



```
thanasis@LAPTOP: ~/Downloads
Four-In-A-Row Server

Connect to one of these addresses:

    lo address: 127.0.0.1
    enp2s0 address: 10.42.0.1
    wlp3s0 address: 192.168.1.13

Please specify the port you want the server to use.
The port number must be between 1025 and 65535.
> 2555

Clients must connect to port 2555
Let the games begin!
Waiting For Player 1...
```

Εφαρμογή Client

Η εφαρμογή του client είναι υπεύθυνη για την διεξαγωγή του παιχνιδιού και την αλληλεπίδραση του χρήστη με το αντίπαλό του καθώς επίσης αποκρύπτει την ύπαρξη του server και από τους δύο χρήστες.

Στάδιο εκκίνησης

Η Εφαρμογή του client εκκινείται από την κονσόλα με παραμέτρους την διεύθυνση ip του διακομιστή και την θύρα στην οποία θα συνδεθεί. Η εφαρμογή κατά την εκκίνηση προσπαθεί να συνδεθεί στο διακομιστή και αν αποτύχει τρεις συνεχόμενες φορές τότε τερματίζει η λειτουργία της. Μετά την επιτυχή σύνδεση ζητά

Αναλυτική Περιγραφή

από το χρήστη να εισάγει το όνομά του και αναμένει κατάλληλο μήνυμα από το διακομιστή ώστε να ξεκινήσει το παιχνίδι.

Στάδιο παιχνιδιού

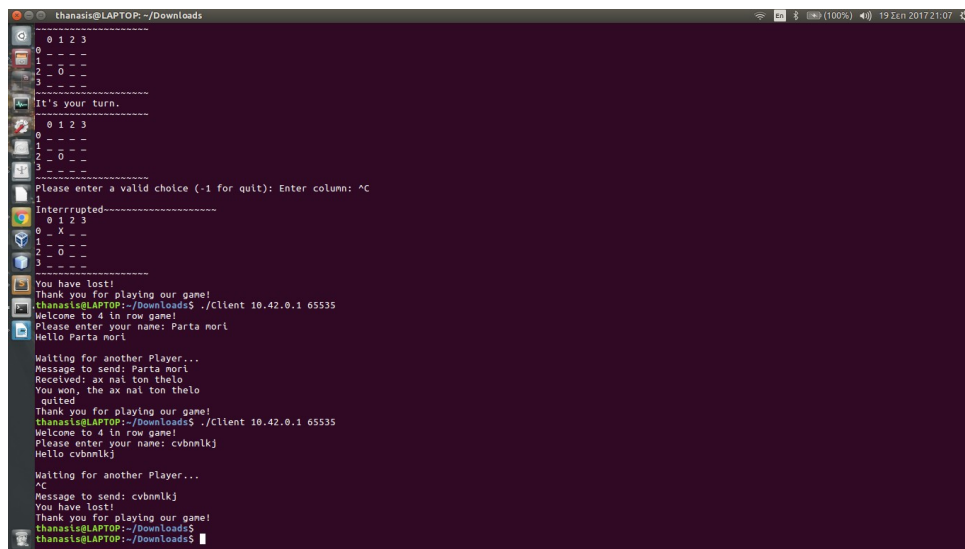
Κατά την διεξαγωγή του παιχνιδιού η εφαρμογή αναμένει το μήνυμα από το διακομιστή. Ανάλογα με το μήνυμα η εφαρμογή αναγνωρίζει τι πρέπει να κάνει, για το αν πρέπει να στείλει κάτι στο διακομιστή ή να περιμένει ένα ακόμα μήνυμα. Τα μηνύματα που λαμβάνει η εφαρμογή πελάτη είναι αντίστοιχα με τα μηνύματα του διακομιστή όπου αναφέρθηκαν πιο πάνω. Η εφαρμογή σε περίπτωση που τερματιστεί με μη κανονικό τρόπο, όπως για παράδειγμα με μια διακοπή του σήματος SIGINT τότε θα σταλεί πρώτα το κατάλληλο μήνυμα στο διακομιστή, όπου είναι το “8”, σηματοδοτώντας ότι ο παίκτης έχει κλείσει το πρόγραμμα και έπειτα τερματίζεται η εφαρμογή [2].

Στην υλοποίησή της η εφαρμογή χρησιμοποιεί ένα νήμα, το οποίο αλληλεπιδρά με το χρήστη και χρησιμοποιείται για την δικτύωση μεταξύ client και server. Η κάθε συνάρτηση ελέγχει μια μεταβλητή-σημαία μέσω της οποίας η εφαρμογή βλέπει αν όλα εκτελούνται σωστά. Σε περίπτωση που το πρόγραμμα συναντήσει κάποιο σφάλμα που δεν μπορεί να διαχειριστεί τότε θα τερματιστεί με κατάλληλο μήνυμα λάθους και θα τερματιστεί η σύνδεση με το διακομιστή με σωστό τρόπο.

Στάδιο τερματισμού

Αναλυτική Περιγραφή

Κατά τον τερματισμό της εφαρμογής το πρόγραμμα ελέγχει μέσω της μεταβλητής-σημαίας αν υπήρξε κάποιο λάθος κατά την διάρκεια εκτέλεσης της εφαρμογής. Αν η μεταβλητή-σημαία έχει την σωστή τιμή τότε γίνεται έλεγχος ώστε να τυπωθεί το σωστό μήνυμα στο χρήστη. Όπως προαναφέρθηκε σε περίπτωση που η εφαρμογή λάβει σήμα το οποίο μπορεί να χειριστεί τότε κλείνει την σύνδεση με σωστό τρόπο και έπειτα τερματίζει ομαλά την λειτουργία της.

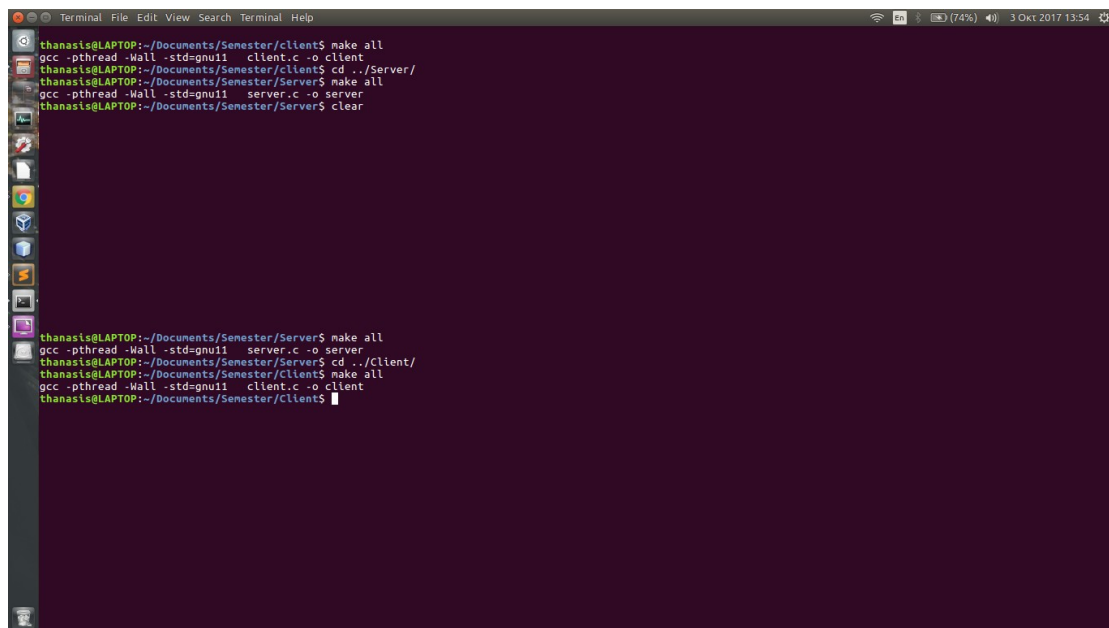


```
thanasislaptop:~/Downloads$
0 1 2 3
0 - - -
1 - - -
2 - 0 -
3 - - -
It's your turn.
0 1 2 3
0 - - -
1 - - -
2 - 0 -
3 - - -
Please enter a valid choice (-1 for quit): Enter column: ^C
Interrupted-----
0 1 2 3
0 - X -
1 - - -
2 - 0 -
3 - - -
You have lost!
Thank you for playing our game!
thanasislaptop:~/Downloads$ ./client 10.42.0.1 65535
Welcome to 4 in row game!
Please enter your name: Parta mori
Hello Parta mori

Waiting for another Player...
Message to send: Parta mori
Received: ax nai ton thelo
You won, the ax nai ton thelo
quited
Thank you for playing our game!
thanasislaptop:~/Downloads$ ./client 10.42.0.1 65535
Welcome to 4 in row game!
Please enter your name: cvbnnlkj
Hello cvbnnlkj

Waiting for another Player...
^C
Message to send: cvbnnlkj
You have lost!
Thank you for playing our game!
thanasislaptop:~/Downloads$
thanasislaptop:~/Downloads$
```


Αναλυτική Περιγραφή



```
thanasis@LAPTOP:~/Documents/Semester/client$ make all
gcc -pthread -Wall -std=gnu11 client.c -o client
thanasis@LAPTOP:~/Documents/Semester/client$ cd ../Server/
thanasis@LAPTOP:~/Documents/Semester/Server$ make all
gcc -pthread -Wall -std=gnu11 server.c -o server
thanasis@LAPTOP:~/Documents/Semester/Server$ clear

thanasis@LAPTOP:~/Documents/Semester/Server$ make all
gcc -pthread -Wall -std=gnu11 server.c -o server
thanasis@LAPTOP:~/Documents/Semester/Server$ cd ../Client/
thanasis@LAPTOP:~/Documents/Semester/Client$ make all
gcc -pthread -Wall -std=gnu11 client.c -o client
thanasis@LAPTOP:~/Documents/Semester/Client$
```

Επιτυχής μεταγλώτιση των αρχείων χωρίς σφάλματα.

Συμπεράσματα και Προβλήματα

Κύριο πρόβλημα αποτέλεσε η αρχική σκέψη που είχαμε η οποία ήταν να γίνονται όλα σειριακά χωρίς όμως να είχαμε λάβει υπόψιν τις διακοπές και κατά συνέπεια τον αποσυγχρονισμό που πιθανόν να υπάρχει. Το πρόβλημα αυτό λύθηκε με τη χρήση των μεταβλητών-σημαιών οι οποίες βοηθούν τόσο την εφαρμογή του διακομιστή όσο και του πελάτη στο να εκτελούν τα σωστά κομμάτια κώδικα. Επίσης πρόβλημα αποτέλεσαν οι συναρτήσεις διαβάσματος από την κονσόλα (`scanf`) και εκτύπωσης (`printf`). Το πρόβλημα λύθηκε με χρήση της συνάρτησης `__fpurge()` που όμως υπάρχει μόνο για Debian εκδόσεις Linux[3].

Η εργασία αυτή συνέβαλε στην καλύτερη κατανόηση της δικτύωσης μεταξύ υπολογιστών καθώς και στην κατανόηση της φιλοσοφίας μοντέλου πελάτη - διακομιστή. Επίσης έγινε κατανοητός ο λόγος για τον οποίο χρησιμοποιούνται τα νήματα στην δικύωση σε τόσο μεγάλο βαθμό, αφού στην τωρινή της έκδοση η εργασία δεν κάνει χρήση των νημάτων, με σκοπό η πολυπλοκότητά της να ανέβηκε χωρίς ιδιαίτερο λόγο.

Βιβλιογραφία

[1] Andrew S. Tanenbaum – Δίκτυα Υπολογιστών (isbn: 960-209-689-6)

[2] <http://man7.org/linux/man-pages/man7/signal.7.html>

[3] <http://manpages.ubuntu.com/manpages/zesty/man3/fpurge.3.html>