Εργαλεία Ανάπτυξης Λογισμικού και Προγραμματισμός Συστημάτων (ΠΛΗ211)

Αναφορά Project 3
Απομακρυσμένη Εκτέλεση Εντολών

Μονομελής Ομάδα Εργασίας :
Μαραγκουδάκης Αντώνιος Εμμανουήλ Α.Μ : 2013030093

Μονομελής ομάδα. Το πρότζεκτ το έκανα μόνος μου. Δούλευα εντός των εορτών και μη έχοντας επαρκή χρόνο δεν υλοποίησα όλους του ελέγχους όπου ζητούσε η εκφώνηση. Αναφέρεται ότι όποιοι δεν ακολουθήσουν τις οδηγίες θα έχουν σημαντική ποινή στη βαθμολογία. Ακολούθησα όσες μπορούσα χρονικά και οι υπόλοιπες δεν ελέγχονται.

Οδηγίες που Ακολουθήθηκαν

Δημιουργίται TCP σύνδεση από τον Client στον Server. Ο Client στέλνει μέσω τις σύνδεσης αυτής 10 – 10 τις εντολές περιμένοντας 5 δευτερόλεπτα στο ενδιάμεσο. Ο Server επιστρέφει το αποτέλεσμα στον σωστό Client που ρώτησε την εντολή, με χρήση τηλεγραφικών υποδοχών (datagram sockets). Ο Client δημιουργεί αρχεία με σωστό όνομα και περιεχόμενο. Ο Server μπορεί να εξυπηρετεί ταυτόχρονα πολλούς Client. Περνάνε μόνο οι 5 ζητούμενες εντολές ενώ οι υπόλοιπες δημιουργούν κενό αρχείο. Ο Server γεννά των σωστό αριθμό παιδιών και η πληροφορία μεταβαίνει από τον πατέρα σε αυτά σωστά μέσω pipes. Ο πατέρας “ακούει” τους Clients, ενώ τα παιδιά “απαντούν” σε αυτούς. Ο Server μπορεί να εξυπηρετεί ταυτόχρονα πολλούς Client. (Multi-Client)

Οδηγίες που δεν Ακολουθήθηκαν

Ο Server δεν στέλνει πακέτα των 512 Bytes αλλά στέλνει όλη την πληροφορία μαζεμένη. Ο Server δεν τερματίζει ποτέ την διεργασία κάποιου παιδιού. Δεν γίνεται χρήση της συνάρτησης select ().

Ζητούμενο του Πρότζεκτ

Ζητούμενο του Πρότζεκτ ήταν να υλοποιήσουμε δύο πρόγραμμα σε C. Το remoteServer.c που θα εκτελεί 5 εντολές UNIX που θα του στέλνει μέσω TCP σύνδεσης το άλλο πρόγραμμα με όνομα remoteClient.c που επίσης υλοποιήθηκε. Ο server θα μπορεί να επικοινωνεί ταυτόχρονα με πολλούς Client και θα τους ξεχωρίζει μέσω της θύρας επιστροφής της απάντησης (recievePort). Με χρήση του remoteClient.c , κάθε Client θα διαβάζει από ένα file εντολές και θα τις στέλνει στον server μέσω TCP σύνδεσης. Για κάθε εντολή θα απαντάει ο server το αποτέλεσμα τις εκτέλεσής τις, στην σωστή recievePort με χρήση τηλεγραφικών υποδοχών. Έπειτα αφού ο κάθε Client πάρει απάντηση μιας εντολής για μία σειρά θα δημιουργήσει ένα αρχείο για κάθε εντολή, με όνομα output.recievePort.Seira. Έτσι ξέρουμε το output σε ποιανού Client είναι και ποιας σειράς η εντολή περιέχεται εκτελεσμένη στο αρχείο αυτό.

Remote Client

Το πρόγραμμα remoteClient.c δέχεται ως ορίσματα το όνομα του μηχανήματος που θα τρέχει τον Server, το serverPORT στο οποίο θα γίνει η TCP σύνδεση με τον Server, το recievePort στο οποίο θα περιμένει απαντήσεις ο Client, καθώς και το όνομα του αρχείο από το οποίο θα διαβάζει τις εντολές που θα στέλνει στον Server.

Προκείμενου ο Client να μπορεί να στέλνει αλλά και να λαμβάνει μηνύματα ταυτόχρονα, χρειάστηκε παραλληλισμός έτσι με την χρήση τις εντολής fork() " γεννήθηκε ένα παιδί " το οποίο θα αναλάβει την διεργασία λήψης μηνυμάτων από τον Server, ενώ ο πατέρας αναλαμβάνει την διεργασία αποστολής μηνυμάτων στον Server.

Η διεργασία του πατέρα αφού έχει συνδεθεί επιτυχώς με τον Server(TCP), στέλνει τις εντολές με την ακολουθία που ζητήθηκε, δηλαδή αποστολή 10 μηνυμάτων, παύση 5 δευτερολέπτων και συνεχίζει ομοίως έως ότου να σταλθούν όλες οι εντολές που διαβάζει από το αρχείο.

Η πληροφορία που στέλνει στον Server για κάθε εντολή-σειρά είναι ένας buffer με περιεχόμενο το **recievePort.σειρά$εντολή** , ετσι ο Server θα ξέρει τι να εκτελέσει, σε ποιόν Client να απαντήσει, και ποιας σειράς (του αρχείου που διαβάζει ο Client) απάντηση δίνει.

Δεν υλοποιήθηκε τερματισμός της λειτουργίας του.

Η διεργασία του παιδιού λαμβάνει τις απαντήσεις του server και όπως και αναφέρθηκε δημιουργεί ένα αρχείο για κάθε εντολή, με όνομα output.recievePort.Seira . Σε περίπτωση μη υποστηριζόμενης εντολής το εκάστοτε αρχείο παραμένει κενό.

Remote Server

Το πρόγραμμα remoteServer.c δέχεται ως ορίσματα το serverPORT στο οποίο θα γίνει η TCP σύνδεση από την οποία θα περιμένει συνδέσεις με τους Clients. Δεύτερο όρισμα παίρνει τον αριθμό numChildren δηλαδή τον αριθμό των "παιδιών που θα γεννήσει" προκειμένου να μπορεί να εξυπηρετεί τους Clients.

Η διεργασία του πατέρα του Server δημιουργεί ένα streaming socket με PORT αυτό που του δόθηκε ως όρισμα (serverPORT). Έτσι μπορεί να ακούει για TCP συνδέσεις από τους πελάτες.

Δέχεται έτσι την πληροφορία που του περνάει ο κάθε Client μέσω του socket αυτού και στην συνέχεια δημιουργεί ένα pipe μέσω του οποίου διοχετεύει την πληροφορία αυτή στα παιδιά που γεννά numChildren φορές, με την χρήση της εντολής fork().

Η διεργασία των παιδιών διαβάζει την πληροφορία όπου έχει διοχετεύσει ο πατέρας σε αυτά μέσω pipes. Η πληροφορία αυτή έχει το φορμάτ **recievePort.σειρά$εντολη** και έτσι κατασκευάζετε και η ανάλογη πληροφορία όπου θα στείλουν οι διεργασίες των παιδιών στους Client.

Έτσι λοιπόν με την χρήση της εντολής popen εκτελείται η κάθε εντολή και δημιουργείται ένας buffer που περιέχει την πληροφορία **recievePort.σειρά$εκτελέσιμο_της_εντολης.**

Ο buffer που μόλις περιγράφτηκε έχει δηλωμένη χωρητικότητα 1024 bytes και στέλνει μονοκόματα όλο την πληροφορία **recievePort.σειρά$εκτελέσιμο_της_εντολης** που θα πεί ότι για να σταλθεί περισσότερη από τόση θα πρέπει να αυξηθεί το μέγεθός του κατά την δήλωση του όσο χρειάζεται. Γενικότερα όλοι οι buffers και temp_buffers που χρησιμοποιήθηκαν έχουν 1024 bytes χωρητικότητα η οποία θα πρέπει ομοίως να αυξηθεί.

Έτσι στην συνέχεια δημιουργεί ένα datagram socket για κάθε απάντηση σε κάθε Client. Στο socket αυτό συνδέεται στο **recievePort** και με αυτόν τον τρόπο στέλνει την πληροφορία του buffer στον κατάληλο Client. Η σειρά της εντολής στέλνεται Client έτσι ώστε αυτός να γνωρίζει ο σε ποια εντολή αναφέρεται η απάντηση έτσι ώστε να δώσει την κατάλληλη ονομασία output.recievePort.Seira στο αρχείο όπου θα δημιουργήσει.

Αν μία εντολή δεν ήταν μία από της ζητούμενες 5 εντολές τότε το μήνυμα που θα σταλεί στον Client θα έχει την μορφή **recievePort.σειρά$** και δεν θα έχει γίνει χρήση της εντολής popen(). Έτσι στην περίπτωση αυτήν ο Client θα σημιουργίσει ένα κενό αρχείο με σωστό όνομα.
Σε περίπτωση σωστής εντολής που έχει κάποιο error κατά την εκτέλεσή της τότε το error αυτό μεταβαίνει στον Client ο οποίος το γράφει στο ανάλογο αρχείο.

Υπάρχει λεπτομερής σχολιασμός και στα δύο αρχεία κώδικα που εξηγεί πιο αναλυτικά την λειτουργία τους.

Παρατηρήσεις

Παρατηρήθηκε στην υλοποίηση αυτή ότι μη τερματίζοντας τις διεργασίες των Client, ο Server μπορούσε να εξυπηρετεί μέχρι numChildren αριθμό από Clients. Θα μπορούσε κάθε Client να τερματίζει μόλις έχει δημιουργήσει τόσα αρχεία όσες εντολές έστειλε στον Server και αυτό ίσως έλυνε το πρόβλημα αλλά αυτήν την υλοποίηση δεν προλαβαίνω να την κάνω καθώς σε μία ώρα η προθεσμία παράδοσης λήγει. **:)**