

Εργαλεία Ανάπτυξης Λογισμικού και Προγραμματισμού Συστημάτων

Αναφορά 3ης άσκησης

Βασίλειος Λυμπεράκης 2015030183

Παναγιώτης Γιαννακόπουλος 2015030171

Ομάδα LAB21142559

Στην εργασία αυτή υλοποιήθηκε η λειτουργία μίας σύνδεσης πελάτη-εξυπηρετητή. Με αυτή την επικοινωνία, ο πελάτης στέλνει Unix εντολές στον εξυπηρετητή και αυτός τις υλοποιεί και στέλνει πίσω το αποτέλεσμα. Στη συνέχεια, ο πελάτης αποθηκεύει τα αποτελέσματα σε αρχεία.

Επικοινωνία Server-Client (Λυμπεράκης Βασίλειος & Γιαννακόπουλος Παναγιώτης)

Για να επιτευχθεί αξιόπιστη επικοινωνία, εισάγεται ένα overhead στο μήνυμα μας. Ο server πριν την αποστολή των πακέτων τα μετατρέπει σε κατάλληλη μορφή και ο client με την λήψη του πακέτου εξάγει την πληροφορία.

Αναλυτικά, τα πακέτα που στέλνει ο server στον client έχουν την εξής μορφή: ID # Index # Response όπου το ID ομαδοποιεί όλα τα πακέτα που αντιστοιχούν στο ίδιο μήνυμα και το Index (με πρώτο στοιχείο το μηδέν) ορίζει την αρίθμηση του πακέτου. Στην περίπτωση που το πακέτο είναι το τελικό, η δεύτερη δίεση (#) αντικαθίσταται από θαυμαστικό (!).

Remote Client (Λυμπεράκης Βασίλειος)

Εκτός της σύνδεσης με τον σέρβερ, στο αρχείο remoteClient εκτελείται το διάβασμα του αρχείου με τις εντολές, η αποστολή τους στον server και η παραλαβή του αποτελέσματος, ώστε να γίνει η καταχώρησή του σε αρχεία.

Αρχικά, διαβάζεται από την είσοδο το όνομα του αρχείου που έχει τις εντολές που θα σταλθούν στον server, εφόσον αυτό είναι εφικτό. Καθώς διαβάζεται το αρχείο ανά γραμμή, μέσω της tcp σύνδεσης στέλνεται η κάθε γραμμή στον server. Για κάθε δέκα εντολές, το διάβασμα διακόπτεται για πέντε δευτερόλεπτα και συνεχίζεται μετά. Ταυτόχρονα με την διαδικασία αυτή εκτελείται και το διάβασμα των πακέτων που προορίζονται για τον client. Αυτό επιτυγχάνεται με τη κλήση της συνάρτησης fork() και τη δημιουργία ενός παιδιού. Με αυτό τον τρόπο, παράλληλα με την αποστολή των εντολών από τον πατέρα, το παιδί λαμβάνει τις απαντήσεις τους.

Όταν εκτελείται το παιδί, ξεκινάει μία επανάληψη, στην οποία με την udp σύνδεση λαμβάνονται οι απαντήσεις από τον server. Κάθε φορά που μία απάντηση καταφθάνει στον client, γίνονται οι κατάλληλες διαδικασίες για να αποθηκευτεί η απάντηση στο σωστό αρχείο. Μέσω ενός ID, τα πακέτα καταχωρούνται στο κατάλληλο αρχείο και μέσω του αναγνωριστικού για το τελικό πακέτο μετρούνται πόσες εντολές έχουν παραληφθεί. Στην περίπτωση που μία εντολή αποτελείται από πολλά πακέτα, τότε αυτά αποθηκεύονται προσωρινά στη μνήμη και περιμένουν να ληφθεί το τελικό πακέτο. Όταν αυτό

ληφθεί ελέγχεται ότι όλα τα απαραίτητα πακέτα για τη συμπλήρωση της απάντησης έχουν ληφθεί και ολοκληρώνεται η δημιουργία του αρχείου. Αν δεν έχουν ληφθεί όλα τα πακέτα αποθηκεύεται και το τελικό πακέτο στη μνήμη και περιμένει την ολοκλήρωση των υπολοίπων. Μόλις συμπληρωθεί ο επιθυμητός αριθμός εντολών και ολοκληρωθούν οι διαδικασίες αποθήκευσης των δεδομένων τους, το παιδί στέλνει ένα σήμα στον πατέρα ότι είναι έτοιμο να τερματίσει. Ο πατέρας με τη σειρά του τερματίζει όλες τις διεργασίες του προγράμματος.

Remote Server (Γιαννακόπουλος Παναγιώτης)

Κατά την εκκίνηση του server δημιουργείται ένα pipe και παράγεται ο δεδομένος αριθμός παιδιών. Για κάθε παιδί κρατάμε το PID του και αν είναι απασχολημένο ή όχι. Ο πατέρας και τα παιδιά από εδώ και στο εξής εκτελούν διαφορετικές λειτουργίες και επικοινωνούν μεταξύ τους μέσω του pipe και των signals.

Λειτουργία πατέρα

Αρχικά ο πατέρας κάνει όλες τις απαραίτητες ενέργειες έτσι ώστε να μπορεί να δέχεται συνδέσεις TCP στο δεδομένα port.

Στην συνέχεια μέσα σε ατέρμον βρόχο ελέγχει μέσω της συνάρτησης select και ενός file descriptor set αν υπάρχουν νέα δεδομένα σε κάποιον file descriptor (fd) όπου διακρίνουμε τις εξής περιπτώσεις:

- Δεδομένα στον FD που ακούμε για νέες συνδέσεις: υπάρχει νέος client, οπότε δημιουργείται ένα νέο socket για τον συγκεκριμένο client μέσω accept και εισάγεται στο FD set.
- Δεδομένα σε άλλο FD: σε αυτή την περίπτωση έχουμε εντολή προς εκτέλεση από κάποιον συνδεδεμένο πελάτη, οπότε εξάγουμε την IP και το port του client και προωθούμε το μήνυμα σε κάποιο από τα παιδιά προς εκτέλεση. Μόλις λάβουμε End of file από τον client τότε ο server κλείνει στην σύνδεση με τον συγκεκριμένο πελάτη.

Πρωώθηση εντολών στα παιδιά

Αναλυτικά για την κατανομή των εντολών στα παιδιά και για τον τρόπο συγχρονισμού ακολουθείται η εξής διαδικασία:

1. Έλεγχος αν η εντολή είναι μικρότερη των 100 χαρακτήρων.
2. Δημιουργία ενός message buffer όπου εισάγονται όλες οι απαραίτητες πληροφορίες προκειμένου το παιδί να είναι σε θέση να απαντήσει στον πελάτη. Συγκεκριμένα το μήνυμα που στέλνεται στο παιδί έχει την εξής μορφή: cmd_size # IP # PORT # Command, όπου cmd_size ο αριθμός των χαρακτήρων της εντολής και τα IP και PORT τα στοιχεία του client.
3. Γράφουμε τα δεδομένα του message buffer στο pipe.
4. Στην συνέχεια επιλέγουμε το πρώτο παιδί που είναι ελεύθερο μέσω του πίνακα child_weight. Αν δεν υπάρχει διαθέσιμο παιδί, περιμένουμε μέχρι να υπάρξει κάποιο. Μόλις βρεθεί κάποιο παιδί, το δηλώνουμε ως απασχολημένο και του στέλνουμε ένα σήμα έτσι ώστε να διαβάσει τα περιεχόμενα του pipe. Επίσης δηλώνουμε το pipe ως δεσμευμένο. Σε αυτό το σημείο στο pipe

δεν υπάρχει άλλη εγγραφή μέχρι να μας ειδοποιήσει με σήμα το παιδί ότι ολοκλήρωσε την λήψη των δεδομένων προκειμένου να μπορεί να ξανά χρησιμοποιηθεί το pipe από τον πατέρα.

5. Δηλώνουμε κάποιο παιδί ως ελεύθερο ξανά όταν μας ειδοποιήσει με σήμα.

Τερματισμός λειτουργίας server

Μόλις ο πατέρας λάβει σήμα για interrupt από κάποιο από τα παιδιά του, ειδοποιεί με την σειρά του τα υπόλοιπα παιδιά να τερματίσουν, περιμένει να ελευθερώσουν τους πόρους τους με την waitpid, εκτυπώνει μήνυμα τερματισμού στο stderr και στο τέλος ολοκληρώνει την λειτουργία του. Ο server επίσης μπορεί να τερματίσει την λειτουργία του όταν δεν υπάρχουν πλέον ενεργά παιδιά λόγω τερματισμού τους μέσω της εντολής «end» από τον client. Ο server μπορεί να τερματιστεί και με το σήμα CTL-C μέσω του terminal.

Λειτουργία κάθε παιδιού

Κατά την εκκίνηση του παιδιού, πραγματοποιείται ο ορισμός των signal handlers και δημιουργείται το socket με πρωτόκολλο UDP για την αποστολή των απαντήσεων στον client. Το παιδί στέλνει τις απαντήσεις στην μορφή που περιεγράφηκε παραπάνω.

Τα βήματα που ακολουθεί το κάθε παιδί σε ατέρμον βρόχο προκειμένου να εξυπηρετήσει τον πελάτη είναι τα εξής:

1. Αρχικά περιμένει να λάβει σήμα από τον πατέρα για να ξεκινήσει να διαβάζει από το pipe.
2. Μόλις λάβει το σήμα, διαβάζει το μήνυμα, εξάγει την IP, το port και την εντολή προς εκτέλεση και ειδοποιεί τον πατέρα ότι έχει τελειώσει την ανάγνωση από το pipe.
3. Στην συνέχεια, αναγνωρίζει την εντολή και αφαιρεί τυχόν τμήματα της που πρέπει να αγνοηθούν σύμφωνα με τις προδιαγραφές της εκφώνησης. Διακρίνουμε τις εξής περιπτώσεις για τις εντολές που λαμβάνονται:
 - a. ls, cat, cut, tr ή grep: εκτέλεση της εντολής με την popen και αποστολή απάντησης στον client.
 - b. end: αποδέσμευση πόρων και τερματισμός λειτουργίας του παιδιού. Ο πατέρας ενημερώνεται με το σήμα SIGCHLD.
 - c. timeToStop: αποστολή σήματος SIGINT στον πατέρα
 - d. Σε διαφορετική περίπτωση στέλνεται στον client το μήνυμα "Invalid Command."
4. Με την ολοκλήρωση της αποστολής της απάντησης στον client το παιδί στέλνει σήμα στον πατέρα και τον ενημερώνει ότι είναι διαθέσιμο για νέες εντολές.

Screenshots

Server

```
panagiotis@panagiotis-PC: ~/Documents/PLH211/Project/Phase 3
File Edit View Search Terminal Help
panagiotis@panagiotis-PC:~/Documents/PLH211/Project/Phase 3$ ./remoteServer 9014 5
Listening for connections to port 9014
connection established with IP : 127.0.0.1 and PORT : 9055
Invalid command: rm abcd
Connection with client with IP: 127.0.0.1 and Port: 9055 CLOSED!
^CChild with pid: 9029 is closing
Child with pid: 9031 is closing
Child with pid: 9030 is closing
Child with pid: 9028 is closing
Child with pid: 9032 is closing
Parent with pid: 9027 is closing
panagiotis@panagiotis-PC:~/Documents/PLH211/Project/Phase 3$
```

Client

```
panagiotis@panagiotis-PC: ~/Documents/PLH211/Project/Phase 3
File Edit View Search Terminal Help
panagiotis@panagiotis-PC:~/Documents/PLH211/Project/Phase 3$ ./remoteClient localhost
9014 9055 inputFileWithCommands
Socket: 5
Server name: localhost
Connecting to localhost port 9014
panagiotis@panagiotis-PC:~/Documents/PLH211/Project/Phase 3$ rm output.9055.
panagiotis@panagiotis-PC:~/Documents/PLH211/Project/Phase 3$ cat inputFileWithCommands
ls
ls; tr
cat README.md
ls
cat remoteServer.c | grep "some"
cat remoteClient.c | tr 'p' 'P'
rm abcd
cat README.md
panagiotis@panagiotis-PC:~/Documents/PLH211/Project/Phase 3$
```