

Άσκηση 3 (Παράδοση μέχρι 23:55, 10/1/2020)

Συνολικό Βάρος άσκησης: 40% του τελικού βαθμού εργαστηρίου

Απομακρυσμένη Εκτέλεση Εντολών.

Στόχος: Εξοικείωση με χρήση υποδοχών (sockets) και σωλήνων (pipes).

Γλώσσα Υλοποίησης: C.

Σημείωση: Εργασίες σε άλλες γλώσσες προγραμματισμού (C++, Java, Python, κτλ) θα μηδενίζονται.

Περιγραφή Λειτουργίας Εξυπηρετητή

Το πρώτο πρόγραμμα, με το όνομα **remoteServer** θα εκτελεί ένα περιορισμένο υποσύνολο από εντολές του Unix που θα του στέλνουν οι πελάτες. Οι εντολές αυτές θα είναι οι εντολές ls, cat, cut, grep και tr. Όλες αυτές οι εντολές μπορεί να περιέχουν παραμέτρους (πχ: ls -alR /usr/) και σωληνώσεις, αρκεί η πρώτη εντολή κάθε σωληνώσεως να είναι ανάμεσα σε αυτές τις 5 εντολές (εννοείται ότι μπορούν να υπάρχουν κενά πριν την κάθε εντολή). **Εντολές με πάνω από 100 χαρακτήρες ή μη υλοποιημένες εντολές διαβάζονται αλλά αγνοούνται.**

Σε περίπτωση σωληνώσεων με κάποια διαφορετική εντολή από τις παραπάνω 5, αγνοείται το κομμάτι από τη μη υλοποιημένη εντολή και πέρα. Σε ακολουθίες εντολών με ερωτηματικά εκτελείται μόνο η πρώτη από τις εντολές, αν αυτή είναι ανάμεσα στις 5 εντολές. Αλλιώς επιστρέφεται μήνυμα σφάλματος στον πελάτη. Για παράδειγμα:

Εντολή	ΣΧΟΛΙΑ	Επιστρέφεται στον πελάτη το αποτέλεσμα...
ls -alR /etc/	Σωστή εντολή	ls -alR /etc/
cut -f1,3 koko tr ':' ' '	Σωστή εντολή	cut -f1,3 koko tr ':' ' '
cut -f1,3 koko; rm -i kokoDir	Σύνθετη εντολή	cut -f1,3 koko
rm -i kokoDir; cut -f1,3 koko	Μη υποστηριζόμενη εντολή rm	Μήνυμα σφάλματος
ls -alR /etc/ rm -i	Μη υποστηριζόμενη εντολή rm	ls -alR /etc/

Δομή Εξυπηρετητή

Το πρόγραμμα **remoteServer** παίρνει 2 παραμέτρους. Η πρώτη παράμετρος (PORT) είναι ο αριθμός της θύρας όπου ο εξυπηρετητής θα ακούει για **TCP** συνδέσεις από πελάτες. Η δεύτερη παράμετρος (NUMCHILDREN >= 1) θα δηλώνει πόσα παιδιά θα γεννήσει η κύρια διεργασία για την εξυπηρέτηση των πελατών. Συνεπώς η κλήση του εξυπηρετητή θα είναι:

./remoteServer portNumber numChildren

Όταν ξεκινάει το πρόγραμμα, ο εξυπηρετητής εκτελεί όλες τις διαδικασίες για να δέχεται **TCP** συνδέσεις και γεννάει NUMCHILDREN διεργασίες παιδιά. **Όλες οι διεργασίες του εξυπηρετητή (πατέρας και παιδιά) να αγνοούν το σήμα SIGPIPE.** Όλες οι αιτήσεις για σύνδεση εξυπηρετούνται ΜΟΝΟ από τη διεργασία πατέρα. Όλες οι εντολές προς εκτέλεση από τους πελάτες διαβάζονται από **τη διεργασία πατέρα** (θυμηθείτε ότι αυτές με πάνω από 100 χαρακτήρες διαβάζονται αλλά αγνοούνται). Αυτές οι εντολές μοιράζονται μετά από τον πατέρα στα παιδιά προς εκτέλεση.

Το μοίρασμα αυτό θα πρέπει να γίνεται με τρόπο ώστε να λαμβάνεται υπ όψιν το φορτίο κάθε παιδιού. Ζητούμενο είναι η καταγραφή των ερωτημάτων να γίνεται σε ένα σωλήνα (pipe) προσπελάσιμο από όλα τα παιδιά. Το πρώτο διαθέσιμο παιδί θα εκτελεί πάντα την πρώτη εντολή που θα βρίσκει στο σωλήνα.

Σε αυτή τη διαδικασία είναι **απαραίτητο** να γνωρίζετε πώς συμπεριφέρονται οι συναρτήσεις **read** και **write** σε σωλήνες. Θα χρειαστείτε **σίγουρα** να συγχρονίσετε την ανάγνωση των μηνυμάτων. Αυτός ο συγχρονισμός να γίνει ΜΟΝΟ με χρήση σωλήνων και όχι με άλλους

τρόπους που δεν έχουμε αναφέρει στο μάθημα (κοινή μνήμη, σηματοφορείς, υποδοχές στο πεδίο UNIX κτλ). Θυμηθείτε ότι λειτουργίες μεγέθους μέχρι PIPE_BUF είναι ατομικές. Για μεγαλύτερα μηνύματα αυτό δεν εξασφαλίζεται.

Αφού κάποιο παιδί έχει διαβάσει μία εντολή, την εκτελεί και επιστρέφει το αποτέλεσμα στον πελάτη που ρώτησε την εντολή με χρήση **τηλεγραφικών υποδοχών (datagram sockets)**. Αν η απάντηση είναι μεγάλη (>512 bytes), τότε το παιδί στέλνει την απάντηση σε πολλά πακέτα μεγέθους το πολύ 512 bytes το καθένα.

Πέραν από τις 5 UNIX εντολές, αν η διεργασία παιδί λάβει μία εντολή **end**, τότε ειδοποιεί τον πατέρα του με ένα σήμα (signal), πχ με την εντολή kill, κλείνει ΟΛΟΥΣ τους περιγραφητές του, αποδεδμεύει όλους τους σημαντικούς πόρους της (γιατί τονίζεται αυτό εδώ?) και τερματίζει τη λειτουργία του. Αν η διεργασία παιδί λάβει την εντολή **timeToStop**, τότε ειδοποιεί με κατάλληλο σήμα τον πατέρα του. Ο πατέρας μετά φροντίζει να τερματίσει τη λειτουργία ΟΛΩΝ των παιδιών (αφού αυτά κλείσουν όλους τους περιγραφητές τους) και στη συνέχεια τερματίζει και ο ίδιος τη λειτουργία του, αφού όμως έχουν τερματίσει ΟΛΑ τα παιδιά του. Οποιαδήποτε διεργασία (πατέρας ή παιδί) τερματίζει τη λειτουργία της τυπώνει κατάλληλο μήνυμα που περιλαμβάνει το αναγνωριστικό της (process id) στον περιγραφητή 2 (stderr).

Δομή Πελάτη

Το πρόγραμμα **remoteClient** παίρνει 4 παραμέτρους. Η πρώτη παράμετρος serverName είναι το όνομα του μηχανήματος στο οποίο τρέχει ο εξυπηρετητής. Η δεύτερη παράμετρος serverPORT δηλώνει τον αριθμό της θύρας όπου ο εξυπηρετητής ακούει για **TCP** συνδέσεις από πελάτες. Η τρίτη παράμετρος receivePort δηλώνει τη θύρα όπου ο πελάτης θα αναμένει μηνύματα μέσω τηλεγραφικής υποδοχής (datagram socket) ως απαντήσεις στις εντολές που στέλνει. Η 4^η εντολή είναι το αρχείο από το οποίο διαβάζει τις εντολές που θα στείλει στο server. Συνεπώς η κλήση θα πρέπει να είναι:

./remoteClient serverName serverPORT receivePort inputFileWithCommands

Ο πελάτης στέλνει τα μηνύματα στο server με την ακόλουθη ακολουθία. Στέλνει 10 μηνύματα (εντολές - αν υπάρχουν τόσες στο inputFile) χωρίς καθυστέρηση, το ένα μετά το άλλο. Μετά περιμένει 5 δευτερόλεπτα, και στέλνει τις επόμενες 10 εντολές (αν υπάρχουν), περιμένει 5 δευτερόλεπτα κτλ. **Προφανώς ο πελάτης θα πρέπει να λαμβάνει και μηνύματα την ώρα που περιμένει ή στέλνει. Σκεφτείτε πώς μπορείτε να το πετύχετε αυτό. Πιθανότατα θα χρειαστείτε να έχετε παραλληλισμό και στον πελάτη.**

Όλες οι απαντήσεις θα πρέπει να τυπώνονται σε αρχεία. Το αποτέλεσμα της 1^{ης} σε σειρά εκτέλεσης εντολής θα γράφεται στο αρχείο output.receivePORT.1 (όπου receivePORT είναι η θύρα όπου ο πελάτης θα αναμένει μηνύματα μέσω τηλεγραφικής υποδοχής), της 2^{ης} σε σειρά εντολής στο αρχείο output.receivePORT.2 κτλ. Όσες είναι οι εντολές στο inputFile, τόσα θα είναι και τα αρχεία. Σε περίπτωση λανθασμένης ή μη υποστηριζόμενης εντολής, το αντίστοιχο αρχείο θα πρέπει να υπάρχει και να είναι KENO.

Όλα τα ερωτήματα (εντολές) στέλνονται στον εξυπηρετητή με βάση την TCP σύνδεση (και οι απαντήσεις λαμβάνονται με τηλεγραφική υποδοχή). Συνεπώς διαφορετικές διεργασίες-παιδιά του εξυπηρετητή μπορούν να απαντήσουν σε διαφορετικές εντολές του ίδιου πελάτη. Σκεφτείτε επίσης ότι τα πακέτα που λαμβάνει ο πελάτης με την τηλεγραφική υποδοχή μπορεί να μην είναι με τη σωστή σειρά, καθώς δεν υπάρχει τέτοια εγγύηση από το πρωτόκολλο επικοινωνίας.

Πράγματα προς σκέψη

- Σκεφτείτε τι πληροφορία θα πρέπει να περνάει η διεργασία πατέρα στα παιδιά για κάθε καινούργια εντολή που λαμβάνει.
- Σκεφτείτε πώς θα διαβάζουν συγχρονισμένα τα παιδιά τα μηνύματα του πατέρα τους. Πώς ξέρουν τα παιδιά πόσο μεγάλο είναι το κάθε μήνυμα;
- Σκεφτείτε πώς θα στέλνουν τα παιδιά απαντήσεις στους πελάτες. Πώς ξέρει ο πελάτης ότι

τελείωσε η απάντηση που περίμενε και δεν ακολουθεί και νέο μήνυμα;

- Πώς μπορεί να ξέρει η διεργασία πατέρα του εξυπηρετητή ποιοι πελάτες έχουν στείλει ερωτήματα? Κοιτάζτε ΚΑΛΑ τη συνάρτηση select (man select). Χρησιμοποιεί για να δείτε ποιοι περιγραφητές έχουν διαθέσιμα δεδομένα χωρίς να μπλοκάρει αν κάπου δεν υπάρχουν δεδομένα.

Ουσιαστικά η διαδικασία χρήσης της είναι:

- ο Δημιουργείτε ένα fd_set στο οποίο θέτεται ένα-ένα τα bits που αντιστοιχούν στον περιγραφητή κάθε ήδη αποδεκτής σύνδεσης και στον περιγραφητή όπου αναμένει για συνδέσεις ο εξυπηρετητής (συναρτήσεις FD_ZERO αρχικά, και μετά διαδοχικά FD_SET).
- ο Κλήση της select με κατάλληλη readfds παράμετρο
- ο Αν μετά τη select ο περιγραφητής όπου ο εξυπηρετητής αναμένει συνδέσεις ικανοποιεί την FD_ISSET, τότε υπάρχει νέα αίτηση για σύνδεση.
- ο Αν μετά τη select ο περιγραφητής κάποιας υπάρχουσας σύνδεσης ικανοποιεί την FD_ISSET, τότε υπάρχουν δεδομένα από αυτόν τον πελάτη.

ΟΔΗΓΙΕΣ ΥΠΟΒΟΛΗΣ

A) Δημιουργήστε ομάδα εργασίας στο μάθημα. Έστω **LAB211XXXXX** η ομάδα σας

B) Δημιουργήστε ένα φάκελο με όνομα ίδιο με το όνομα της ομάδας σας (**LAB211XXXXX**)

Γ) Ο φάκελος μέσα θα πρέπει να περιέχει **Makefile** και αναφορά, εκτός από τα αρχεία για τον πελάτη και τον εξυπηρετητή. Με απλή εντολή **make** θα πρέπει να γίνεται η παραγωγή των εκτελέσιμων αρχείων χωρίς κανένα error/warning από τον κώδικά σας. **Τα εκτελέσιμα αρχεία θα πρέπει να έχουν τα ονόματα που αναφέρονται στην εκφώνηση.** Είναι **ΣΗΜΑΝΤΙΚΟ** να είναι με την παραπάνω ονομασία ο φάκελος και τα αρχεία σας.

Δ) Για να δημιουργήσετε το αρχείο υποβολής του project σας κάντε cd στο γονικό φάκελο του project σας (βγείτε έξω από το φάκελο **LAB211XXXXX** του project) και δημιουργήστε ένα συμπίεσμένο αρχείο με την εντολή

tar zcvf LAB211XXXXX.tar.gz LAB211XXXXX

Εργασίες που δεν έχουν ακολουθήσει τις παραπάνω οδηγίες θα έχουν σημαντική ποινή στη βαθμολογία. Το αναφέρω σε κάθε εκφώνηση, αλλά ακόμα και στο 2^ο πρότζεκτ υπήρχαν ομάδες που δεν ακολούθησαν τις οδηγίες και θα έχουν ποινή, όπως είχα αναφέρει.

Τα διαδικαστικά του μαθήματος περιγράφουν τον κώδικα δεοντολογίας. Υπενθυμίζω ότι στην αρχή της αναφοράς σας ΠΡΕΠΕΙ να περιγράψετε το ποιο κομμάτι της άσκησης υλοποίησε το κάθε μέλος της ομάδας. **Η αναφορά βαθμολογείται.**

Καλή επιτυχία!