

Αναφορά ΗΜΥ316

Πανεπιστήμιο Κύπρου

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Assignment 6 (Due 10/03/2022)

Όνομα και Ταυτότητα :

Εντουίνα Κάρουλλα 1042364

Ομάδα : 7

Φοίβος Λύμπουρας	1016477
Στέλιος Καραγιώργης	1021340
Θεοδόσιος Ιωάννου	1020844



Άσκηση 1:

socket (): Παίρνει ως ορίσματα το address domain του socket, τον τύπο του socket καθώς και το πρωτόκολλο που χρησιμοποιεί. Χρησιμοποιείται σαν system call για τη δημιουργία καινούργιου socket (το ένα άκρο στην ενδοεπικοινωνία).

bind (): Με την εντολή αυτή, αναθέτουμε διεύθυνση και το μέγεθος της (bytes) ανάλογα με το είδος της διευθυνσιοδότησης επειδή όταν δημιουργείται ένα socket δεν έχει καθορισμένη διεύθυνση.

listen (): Καθορίζει το όριο της ουράς για τις εισερχόμενες αιτήσεις σύνδεσης. Έχει ως παραμέτρους το socket και το μέγιστο μέγεθος της ουράς με τις αιτήσεις που περιμένουν για σύνδεση.

accept (): Από την ουρά με τις εκκρεμείς συνδέσεις (listen) εξάγει την πρώτη αίτηση για σύνδεση. Δημιουργείται καινούργιο συνδεδεμένο socket και επιστρέφει ένα καινούργιο file που αναφέρεται σε αυτό. Δέχεται ως ορίσματα το socket, το address_length και το socket_address.

connect (): System Call που συνδέει ένα socket με μια διεύθυνση που θα του δοθεί. Για να καθοριστεί ο τύπος της διευθυνσιοδότησης πρέπει να δοθεί και το μέγεθος (bytes) της διεύθυνσης. Με αυτό τον τρόπο πετυχαίνουμε την έναρξη επικοινωνίας. Έχει ως ορίσματα το socket, address_length και serv_addr.

serve(): Με την κλήση της συνάρτησης αυτής διαβάζει το μήνυμα του client και το τυπώνει. Μετά διαβάζει ένα ένα τους χαρακτήρες από το πληκτρολόγιο και σταματά όταν πατηθεί το enter και τέλος στέλνει το μήνυμα από τον server

read (): Προσπαθεί να διαβάσει συγκεκριμένο αριθμό bytes (που προϋποθέτει το count) από file descriptor και τα τοποθετεί σε ένα buffer. Έχει ως ορίσματα το file descriptor, το count και το buf.

write (): Έχει τα ίδια ορίσματα με το read() και γράφει μέχρι count bytes στο file που καθορίζεται από το file descriptor ξεκινώντας από το buf του buffer. Προϋποθέτει να ακολουθήσει read μετά.

close (): Χρησιμοποιείται για να μην γίνονται αναφορές σε κάποιο file ώστε να μπορεί να ξαναχρησιμοποιηθεί κλείνοντας το file descriptor που δέχεται ως όρισμα



Άσκηση 2:

Objective:

1. Δημιουργήστε ένα πρόγραμμα που υλοποιεί το πρωτόκολλο διακοπής και αναμονής.
2. Λάβαμε μια συμβολοσειρά εισόδου από το χρήστη στην υποδοχή του υπολογιστή-πελάτη.
3. Κάθε χαρακτήρας της εισόδου συμβολοσειράς αποστέλλεται στο διακομιστή με τον αριθμό ακολουθίας (0 ή 1 ως char), 2 χαρακτήρες συνολικά για κάθε πακέτο
4. Εμφανίζουμε κάθε πακέτο και επιβεβαίωση που αποστέλλεται και λαμβάνεται στο κανάλι, το διακομιστή και τον υπολογιστή-πελάτη ανάλογα.

Explanation:

1. Χρησιμοποιήσαμε ένα κανάλι επικοινωνίας μεταξύ του υπολογιστή-πελάτη ως διακομιστή ως μέσο για τη μεταφορά των πακέτων από τον υπολογιστή-πελάτη στο διακομιστή.
2. Κάθε πακέτο έχει τη δική του μοναδική ακολουθία που αποκωδικοποιείται σε 0 και 1.
3. Η κύρια λογική είναι ότι το κανάλι είναι τόσο δέκτης όσο και αποστολέας / διακομιστής.
4. Λαμβάνει πακέτα από τον υπολογιστή-πελάτη και τα στέλνει στο διακομιστή.
5. Λαμβάνει αναγνώρισεις από το διακομιστή και τις στέλνει στον υπολογιστή-πελάτη.
6. 2 θύρες απαραίτητες (κανάλι προγράμματος-πελάτη, channel-server).
7. Ορίστε τις αρχικές συνθήκες για την αποστολή του πρώτου πακέτου (επιβεβαίωση που ελήφθη στον υπολογιστή-πελάτη για να ξεκινήσετε την επικοινωνία και να στείλετε πακέτα στο διακομιστή μέσω του καναλιού)
8. Όταν η αναγνώριση για ένα συγκεκριμένο πακέτο αποστέλλεται στον υπολογιστή-πελάτη, τότε ο υπολογιστής-πελάτης συνεχίζει (τερματίζει τη διακοπή και την αναμονή) και προετοιμάζει και στέλνει το επόμενο πακέτο.
9. Εν τω μεταξύ, ο διακομιστής σταματά και περιμένει το επόμενο πακέτο.
10. Το πακέτο που λαμβάνεται από υπολογιστή-πελάτη σε διακομιστή, οι έξοδοι διακοπής και αναμονής του διακομιστή, στέλνουν επιβεβαίωση.
11. Η αναγνώριση που λαμβάνεται από το διακομιστή στον υπολογιστή-πελάτη, οι έξοδοι πελατών σταματούν και περιμένουν να στείλουν το επόμενο πακέτο και ούτω καθεξής.
12. Η εφαρμογή μας δεν αντιμετώπισε απώλειες πακέτων και, ως εκ τούτου, δεν ήταν απαραίτητος χρονομερητής για την απόρριψη και την αποστολή των πακέτων.



```
$ ./client
Enter Data: AAbC2
[+]Frame 0 Send: A
[+]Ack 0 Received
[+]Frame 1 Send: A
[+]Ack 1 Received
[+]Frame 0 Send: b
[+]Ack 0 Received
[+]Frame 1 Send: C
[+]Ack 1 Received
[+]Frame 0 Send: 2
[+]Ack 0 Received
Enter Data: 
```

```
./channel
[+]Frame 0 Received: A
[+]Frame 0 Send: A
[+]Ack 0 Received
[+]Ack 0 Send
[+]Frame 1 Received: A
[+]Frame 1 Send: A
[+]Ack 1 Received
[+]Ack 1 Send
[+]Frame 0 Received: b
[+]Frame 0 Send: b
[+]Ack 0 Received
[+]Ack 0 Send
[+]Frame 1 Received: C
[+]Frame 1 Send: C
[+]Ack 1 Received
[+]Ack 1 Send
[+]Frame 0 Received: 2
[+]Frame 0 Send: 2
[+]Ack 0 Received
[+]Ack 0 Send

```

```
$ ./server
[+]Frame Received 0: A
[+]Ack 0 Send
[+]Frame Received 1: A
[+]Ack 1 Send
[+]Frame Received 0: b
[+]Ack 0 Send
[+]Frame Received 1: C
[+]Ack 1 Send
[+]Frame Received 0: 2
[+]Ack 0 Send

```