

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ  
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ



**ΕΡΓΑΣΤΗΡΙΟ ΔΙΚΤΥΩΝ**

**1<sup>Η</sup> ΑΣΚΗΣΗ**

Φώτιος Διονυσόπουλος ΑΜ:5753

e-mail: [dionys@ceid.upatras.gr](mailto:dionys@ceid.upatras.gr)

Απρίλιος 2017

## Υλοποίηση Client

Υλοποιήθηκε στο αρχείο client.c και δέχεται έναν μεταβλητό αριθμό από args τα οποία θα πρέπει να έχουν συγκεκριμένη μορφή.

Πιο συγκεκριμένα, μέσω της gethostbyname, γίνεται σύνδεση με τον host του command line argument.

Έπειτα, δημιουργείται το socket επικοινωνίας και αρχικοποιούνται κάποιες μεταβλητές.

Ακολουθεί το connect, το οποίο ανοίγει την σύνδεση με τον server.

Το βασικό κομμάτι του client βρίσκεται μέσα στην for που ακολουθεί. Μέσα στην for γίνεται έλεγχος όλων των args και μέσω της Writen, γίνεται η αποστολή των δεδομένων στον server, πάντα τηρώντας το πρωτόκολλο επικοινωνίας.

Πρωτόκολλο: Μια εντολή ξεκινά με ένα byte που ορίζει το είδος της λειτουργίας, το 103 σημαίνει get ενώ το 112 σημαίνει put. Οποιοσδήποτε άλλος κωδικός θεωρείται λάθος και ο server θα πρέπει να τερματίζει τη σύνδεση.

Μόλις εντοπιστεί η λέξη “get” αποστέλλεται ο κωδικός 103(g), ακολουθεί η written η οποία στέλνει το key και μετά μια recn η οποία λαμβάνει το value(εάν υπάρχει). Επίσης, κατά την λήψη των δεδομένων ακολουθεί η συνάρτηση poll με την οποία αν δεν ληφθούν τα δεδομένα μέσα σε κάποιο χρονικό διάστημα τότε προχωρά η εκτέλεση.

Ενώ αν εντοπιστεί η λέξη “put” αποστέλλεται ο κωδικός 112(p), ακολουθεί η αποστολή του key και του Value.

Η Writen πρόκειται για μια συνάρτηση Wrapper η οποία χρησιμεύει στην αποστολή των δεδομένων στον server. Ο κώδικας αυτής (Stevens book, σελίδα 78):

```
ssize_t writen(int fd, const void *vptr, size_t n)
{
    size_t nleft;
    ssize_t nwritten;
    const char *ptr;
    ptr = vptr;
    nleft = n;
    while (nleft > 0) {
        if ((nwritten = write(fd, ptr, nleft)) <= 0) {
            if (errno == EINTR)
                nwritten = 0;
            else
                return -1; /* error */
        }
        nleft -= nwritten;
        ptr += nwritten;
    }
    return n;
}
```

## Iterative Server

Ο πρώτος server που υλοποιήθηκε είναι ο Iterative ο οποίος δέχεται έναν client κάθε φορά.

Η διαδικασία που ακολουθείται για την σύνδεση με τον client είναι η εξής: δημιουργία socket, ανάθεση τιμών σε μεταβλητές(setsockopt), bind, listen με Backlog 5 και Accept.

Η παραπάνω διαδικασία ακολουθείται σε όλους τους server.

Έπειτα, ακολουθεί η While(1) μέσα στην οποία γίνεται έλεγχος (μέσω της recv) για τους επιθυμητούς κωδικούς πρωτοκόλλου. Η recv λαμβάνει το string χαρακτήρα-χαρακτήρα προκειμένου να σταματήσει όταν εντοπίσει το τερματικό '\0', έτσι ώστε να μπορεί να ξεχωρίσει τα token που λαμβάνονται. Επίσης, τα δεδομένα που λαμβάνονται αποθηκεύονται μέσα σε έναν buffer του οποίου το μέγεθος αυξάνεται μέσω της realloc.

Όταν εντοπιστεί ο κωδικός για get, χρησιμοποιείται μια συνάρτηση, η search, η οποία ψάχνει στην δομή δεδομένων για το επιθυμητό Value, εάν δεν υπάρχει τότε στέλνεται ο κωδικός "n" (not found) ενώ εάν βρεθεί, στέλνεται ο κωδικός "f" και έπειτα το value για να το εκτυπώσει ο client.

Ενώ για το put, χρησιμοποιείται μια συνάρτηση, η add\_value, η οποία ψάχνει στην δομή δεδομένων αν υπάρχει ο Key, αν υπάρχει το ενημερώνει, αλλιώς θέτει μια νέα τιμή στον πίνακα μαζί με το αντίστοιχο Value.

Δομή δεδομένων: Η δομή δεδομένων αποτελείται από έναν 2D πίνακα από strings.

## On-demand forking server

Σε αυτή την περίπτωση, αφού πρόκειται για επιπλέον διεργασίες, δημιουργήθηκε shared memory στη δομή δεδομένων. Πιο συγκεκριμένα, δημιουργήθηκε ένα struct στο οποίο βρίσκεται ο 2D πίνακας καθώς επίσης και ένας δείκτης στην επόμενη ελεύθερη θέση του πίνακα.

Η βασική αλλαγή είναι η προσαρμογή του server έτσι ώστε η διεργασία παιδί να εκτελεί την μεταφορά – αποστολή των δεδομένων. Επομένως, μετά το Accept της σύνδεσης προστέθηκε το Fork το οποίο δημιουργεί την διεργασία η οποία θα εκτελέσει τις εντολές του client.

## Preforking server

Σε αυτόν τον Server, τα Forking γίνονται από πριν, και μόλις υπάρχει νέα σύνδεση από client τότε κάποια διεργασία κάνει Accept.

Ο αριθμός των διεργασιών που θα δημιουργηθούν περνά σαν Argument από πριν μέσω του Command Line.

Επομένως, δημιουργήθηκε μια επιπλέον While(1) στην οποία κάθε διεργασία περιμένει μέχρι να δοθεί σήμα από κάποιον Client για να τρέξει την Accept.

Ψευδοκώδικας:

```
#define NB_PROC 10
int main() {
    int newfd, fd=socket(AF_INET, SOCK_STREAM, 0); // bind() and listen() have been omitted
    for (int i=0; i<NB_PROC; i++) /* Create NB_PROC children */
        if (fork()==0)
            while (1) {
                newfd = accept(fd,...);
                treat_request(newfd);
                close(newfd); }
}
```

## Multi-Threaded Server

ASD