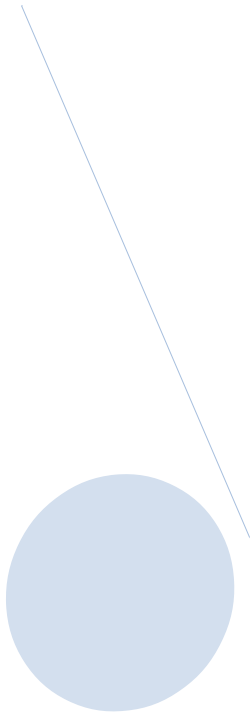




ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ

ATM PROJECT

2011-2012



Άννα Τριανταφύλλου

ΑΕΜ : 456

επιβλέπων καθηγητής: Μηνάς Δασυγένης

Περιεχόμενα

1. Δήλωση καθολικών μεταβλητών
2. Κύριο πρόγραμμα (Server)-Main Function
 - a. Τοπικές μεταβλητές
 - b. Ανάλυση
3. Thread Function(Client)- Request Processing
 - a. Τοπικές μεταβλητές
 - b. Ανάλυση
 - c. Αν ο χρήστης είναι ο πελάτης
 - d. Αν ο χρήστης είναι ο administrator
4. Screenshots
5. Σημείωση
- 6.**

Περίληψη

Το συγκεκριμένο πρόγραμμα αποτελεί την υλοποίηση ενός TCP multithreaded web server στο λειτουργικό σύστημα Ubuntu, του οποίου οι λειτουργίες βασίζονται στο ATM μίας τράπεζας.

Κάθε χρήστης που συνδέεται στο σύστημα έχει τη δυνατότητα να εκτελέσει μία από τις παρακάτω λειτουργίες :

1. Κατάθεση
2. Ανάληψη
3. Ερώτηση Υπολοίπου
4. Ιστορικό Καταθέσεων
5. Ιστορικό Αναλήψεων
6. Έξοδος (Exit)

Ο χρήστης μπορεί να εκτελέσει μία μόνο λειτουργία τη φορά και στο όνομά του είναι καταχωρημένος μόνο ένας λογαριασμός τύπου όψεως. Δηλαδή έχει τη δυνατότητα να τραβήξει χρήματα από την τράπεζα μέχρι ένα συγκεκριμένο όριο και ας έχει μηδενικό υπόλοιπο.

Για να γίνει η εισαγωγή του στο σύστημα θα πρέπει να δώσει το προσωπικό του κωδικό, τον οποίο και θα πληκτρολογεί κάθε φορά που θέλει να εκτελέσει μία διαφορετική λειτουργία από τις παραπάνω.

Για την πραγματοποίηση της αίτησης του κάθε πελάτη χρησιμοποιείται ένα thread. Δηλαδή όσοι θα είναι οι πελάτες που θα βρίσκονται ταυτόχρονα στο σύστημα, τόσα θα είναι και τα threads διότι τόσες λειτουργίες θα πρέπει να εκτελεστούν.

Για να γίνει η εκτέλεση του προγράμματος χρειάζονται δύο τερματικά. Στο ένα από αυτά θα γίνει το compile και η εκτέλεση του προγράμματος. Το τερματικό αυτό θα αποτελεί το server. Ενώ στο δεύτερο τερματικό, το οποίο θα αποτελεί τον client, θα γίνει η εκτέλεση της παρακάτω εντολής

```
telnet localhost 5080
```

έτσι ώστε να πραγματοποιηθεί η σύνδεση του client με το server.

Το πρόγραμμα αποτελείται από 3 μέρη:

- **Δήλωση καθολικών μεταβλητών**
- **Thread Function(Client)- Request_Processing**
- **Κύριο πρόγραμμα (Server)-Main Function**

Η ανάλυση του προγράμματος θα γίνει με βάση τη λειτουργία του, οπότε αρχικά θα αναλυθεί η δήλωση των μεταβλητών, έπειτα το κύριο πρόγραμμα (Server)-Main Function και τέλος η thread function.

1. Δήλωση καθολικών μεταβλητών

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <errno.h>
#include <sys/types.h>
#include <netinet/in.h>
```

```
#include <sys/socket.h>
#include <arpa/inet.h>
#include <pthread.h>
#include <signal.h>
#include <sched.h>
#include <semaphore.h>
```

Δήλωση των απαραίτητων βιβλιοθηκών για την εκτέλεση των υπαρχόντων συναρτήσεων.

- **number_of_clients** : ο αριθμός των πελατών που βρίσκονται καταχωρημένοι στο σύστημα. Για λόγους ευκολίας υπάρχουν μόνο 3 πελάτες καταχωρημένοι.
- **number_of_connections** : ο αριθμός των συνδέσεων που μπορούν να υπάρχουν ταυτόχρονα στο σύστημα, όσοι είναι και οι πελάτες δηλαδή.
- **rows** : μια σταθερά που χρησιμοποιείται για τον ορισμό του μεγέθους ενός πίνακα.

- **columns** : μια σταθερά που χρησιμοποιείται για τον ορισμό του μεγέθους ενός πίνακα.
- **n** : μια σταθερά που χρησιμοποιείται για τον ορισμό του μεγέθους ενός πίνακα.
- **struct clients_accounts** : είναι μια δομή με το όνομα `clients_accounts` η οποία περιέχει τα στοιχεία του κάθε πελάτη. Δηλαδή τι χρειάζεται να ξέρει για αυτόν η τράπεζα.
 - **char *account_number** : είναι μια μεταβλητή δείκτης τύπου `char`, η οποία αποτελεί το κωδικό του πελάτη για την εισαγωγή του στο σύστημα. Το μέγεθός του εξαρτάται από το μέγεθος του κωδικού του πελάτη.
 - **char *first_name** : είναι μια μεταβλητή δείκτης τύπου `char`, η οποία αποτελεί το όνομα του πελάτη.
 - **char *last_name** : είναι μια μεταβλητή δείκτης τύπου `char`, η οποία αποτελεί το επώνυμο του πελάτη.
 - **double *ypoloipo** : είναι μια μεταβλητή δείκτης τύπου `double`, η οποία αποτελεί έναν πίνακα με το υπόλοιπο του λογαριασμού του κάθε πελάτη. Το μέγεθός του εξαρτάται από το πόσες φορές αλλάξει το υπόλοιπο του πελάτη.
 - **double *pinakas_katathesewn** : είναι μια μεταβλητή δείκτης τύπου `double`, η οποία αποτελεί ένα πίνακα, ο οποίος περιέχει τις καταθέσεις του κάθε πελάτη. Το μέγεθός του εξαρτάται από το πόσες καταθέσεις θα κάνει ο πελάτης.
 - **double *pinakas_analypsewn** : είναι μια μεταβλητή δείκτης τύπου `double`, η οποία αποτελεί ένα πίνακα, ο οποίος περιέχει τις αναλήψεις του κάθε πελάτη. Το μέγεθός του εξαρτάται από το πόσες αναλήψεις θα κάνει ο πελάτης.
 - **char **pinakas_dates_analypsewn** : είναι μια μεταβλητή δείκτης τύπου `char`, η οποία αποτελεί ένα δυσδιάστατο πίνακα, ο οποίος περιέχει τις ημερομηνίες αναλήψεων του κάθε πελάτη. Σαν αριθμό γραμμών έχει τον αριθμό των αναλήψεων, καθώς κάθε ανάληψη πρέπει να έχει και μια ημερομηνία και σαν αριθμό στηλών έχει το μέγεθος του `string` της ημερομηνίας.
 - **char **pinakas_dates_katathesewn** : είναι μια μεταβλητή δείκτης τύπου `char`, η οποία αποτελεί ένα δυσδιάστατο πίνακα, ο οποίος περιέχει τις ημερομηνίες καταθέσεων του κάθε πελάτη. Σαν αριθμό

γραμμών έχει των αριθμό των καταθέσεων, καθώς κάθε κατάθεση πρέπει να έχει και μια ημερομηνία και σαν αριθμό στηλών έχει το μέγεθος του string της ημερομηνίας.

- **int client_socket** : είναι μια μεταβλητή τύπου int, η οποία αποτελεί το file descriptor του socket του πελάτη για την εισαγωγή του στο σύστημα.
- **int server_socket** : είναι μια μεταβλητή τύπου int, η οποία αποτελεί το file descriptor του socket του διαχειριστή.

2. Κύριο πρόγραμμα (Server)-Main Function

Μέσα στη συνάρτηση main εκτελούνται οι εξής εντολές και βρίσκονται τα εξής δεδομένα :

a. Τοπικές Μεταβλητές

- **int address_size** : μια μεταβλητή τύπου int, η οποία αποτελεί το μέγεθος της διεύθυνσης διαδικτύου
- **struct sockaddr_in server_address** : μια μεταβλητή τύπου struct, η οποία αποτελεί τη διεύθυνση διαδικτύου του διαχειριστή.
- **struct sockaddr_in client_address** : μια μεταβλητή τύπου struct, η οποία αποτελεί τη διεύθυνση διαδικτύου του πελάτη.
- **int id** : μια μεταβλητή τύπου int, η οποία περιλαμβάνει τις default παραμέτρους των threads.
- **pthread_attr_t attr** : μια μεταβλητή τύπου pthread_attr_t, η οποία περιλαμβάνει τα default attributes των threads.
- **pthread_t work_threads** : ο ορισμός των threads, με το όνομα work_threads.

b. Ανάλυση

Αρχικά αρχικοποιείται το αντικείμενο των thread attributes (attr) με default attribute αξίες, μέσω της συνάρτησης **pthread_attr_init()**:

pthread_attr_init(&attr)

και αμέσως μετά δημιουργείται μια καινούργια σύνδεση (socket) με την παρακάτω εντολή :

```
server_socket = socket(AF_INET,SOCK_STREAM,0);
```

όπου η παράμετρος **AF_INET** καθορίζει την περιοχή επικοινωνίας, η οποία στη συγκεκριμένη περίπτωση περιλαμβάνει τα IPv4 πρωτόκολλα διαδικτύου. Η παράμετρος **SOCK_STREAM** που καθορίζει την αμφίδρομη, αξιόπιστη και σε byte stream βασιζόμενη επικοινωνία. Και τέλος το μηδέν που καθορίζει το πρωτόκολλο.

Έπειτα καθορίζονται τα χαρακτηριστικά της διαδικτυακής διεύθυνσης, δηλώνοντας ότι θα χρησιμοποιηθεί η θύρα 5080 για τη σύνδεση

```
server_address.sin_family=AF_INET;  
server_address.sin_port=htons(5080);  
server_address.sin_addr.s_addr=htonl(INADDR_ANY);
```

αλλά και η συνάρτηση

```
bind (server_socket,(struct sockaddr *)&server_address,sizeof(server_address))
```

```
/* όπου struct sockaddr* δείκτης σε μια δομή sockaddr, */
```

η οποία θα συνδέσει τη διαδικτυακή διεύθυνση (**server_address**) με το socket που περιγράφεται από το file descriptor **server_socket** του διαχειριστή.

Στη συνέχεια χρησιμοποιείται η συνάρτηση listen() για την ανίχνευση εισερχόμενων αιτήσεων σύνδεσης με το διαχειριστή, οι οποίες θα γίνουν αποδεκτές ή όχι με τη βοήθεια της συνάρτησης accept().

Μέσα λοιπόν σε έναν ατέρμων βρόγχο :

- μετράτε το μέγεθος της διεύθυνσης και
- με τη χρήση της συνάρτησης

```
client_socket=accept(server_socket,(struct sockaddr*)&client_address,&address_size)
```

```
/* όπου struct sockaddr* δείκτης σε μια δομή sockaddr, */
```

επιστρέφεται ο file descriptor του πελάτη (**client_socket**) μέσω του οποίου θα διαπιστωθεί εάν είναι δυνατό να πραγματοποιηθεί η σύνδεση ή όχι.

- Εάν η σύνδεση δεν μπορεί να πραγματοποιηθεί θα εμφανιστεί μήνυμα λάθους και ο διαχειριστής θα συνεχίσει την ακρόαση. Διαφορετικά θα

δημιουργηθεί ένα thread για την ικανοποίηση της αίτησης του πελάτη μέσω της συνάρτησης

pthread_create(&work_threads ,&attr, request_processing, &id)

όπου η μεταβλητή **attr** δείχνει στην δομή pthread_attr_t, η οποία είναι υπεύθυνη για την επιλογή των attributes των νέων threads και **request_processing** είναι η συνάρτηση επεξεργασίας του κάθε thread.

3. Thread Function (Client) - Request Processing

a. Τοπικές μεταβλητές

int newClient_socket : μια μεταβλητή τύπου int στην οποία θα αντιγράψουμε το file descriptor του πελάτη.

char client_request1[2] : μια μεταβλητή πίνακας τύπου char.

int bytes : μια μεταβλητή τύπου int.

char *client_request=&client_request1[0] : μια μεταβλητή δείκτης τύπου char.

char date[columns] : μια μεταβλητή πίνακας τύπου char.

char *dt=&date[0] : μια μεταβλητή δείκτης τύπου char.

char pin[n] : μια μεταβλητή πίνακας τύπου char.

char *p=&pin[0] : μια μεταβλητή δείκτης τύπου char.

int i,j,k,v,h,m,b,x : μεταβλητές τύπου int.

int counter1=0 : μια μεταβλητή τύπου int.

char new_pin[n] : μια μεταβλητή πίνακας τύπου char.

char *n_p=&new_pin[0] : μια μεταβλητή δείκτης τύπου char.

int thesi_pnks_katathesewn=0 : μια μεταβλητή τύπου int.

int thesi_pnks_analyypsewn=0 : μια μεταβλητή τύπου int.

int thesi_pnks_ypoloipo=0 : μια μεταβλητή τύπου int

char first_n[20] : μια μεταβλητή πίνακας τύπου char.

char *f_n=&first_n[0] : μια μεταβλητή δείκτης τύπου char.

char last_n[20] : μια μεταβλητή πίνακας τύπου char.

char *l_n=&last_n[0] : μια μεταβλητή δείκτης τύπου char.

int client_position : μια μεταβλητή τύπου int.

char use1[15] : μια μεταβλητή πίνακας τύπου char.

char *use=&use1[0] : μια μεταβλητή δείκτης τύπου char.

double amount : μια μεταβλητή τύπου double.

double *temp : μια μεταβλητή δείκτης τύπου double.

char *test3 : μια μεταβλητή δείκτης τύπου char.

int c1=0,c=0 : μεταβλητές τύπου int.

char *msg : μια μεταβλητή δείκτης τύπου char.

char *test : μια μεταβλητή δείκτης τύπου char.

char *dt_print : μια μεταβλητή δείκτης τύπου char.

double *test2 : μια μεταβλητή δείκτης τύπου double.

char temp_9999[n]="9999" : μια μεταβλητή πίνακας τύπου char.

b. Ανάλυση

Αρχικά πραγματοποιείται η αρχικοποίηση του πίνακα **use1** και **use2**, η αντιγραφή του **client_socket** στην τοπική μεταβλητή **newClient_socket** και η δημιουργία ενός μονοδιάστατου δυναμικού πίνακα που θα αποτελείται από δομές **clients_ammounts**. Ο πίνακας αυτός έχει το όνομα **clients** και κάθε κελί του θα περιλαμβάνει τη δομή ενός πελάτη. Δεν γνωρίζουμε τον αριθμό των πελατών εξαρχής και για το λόγο αυτό ο πίνακας είναι δυναμικός. Βέβαια για λόγους ευκολίας στην εκτέλεση του προγράμματος όπως θα δούμε και παρακάτω οι καταχωρημένοι πελάτες θα είναι μόνο τρεις. Έπειτα χρησιμοποιώντας ένα βρόγχο επαναλήψεων δημιουργείται ξεχωριστά χώρος στη μνήμη και για τα περιεχόμενα της κάθε δομής που περιέχει ο πίνακας **clients**. Οι πίνακες **pinakas_katathesewn**, **ypoloipo**, **pinakas_analypsewn**, **pinakas_dates_katathesewn**, **pinakas_dates_analypsewn** για την ευκολότερη εκτέλεση του προγράμματος έχουν οριστεί δυναμικά με μέγεθος `rows=2000` για τους μονοδιάστατους και `rows=2000, columns=11` για τους δισδιάστατους.

Στη συνέχεια αρχικοποιούνται τα στοιχεία των πελατών που θα είναι καταχωρημένοι στη μνήμη κάθε φορά που τρέχει το πρόγραμμα. Οι πελάτες αυτοί θα είναι τρεις και θα έχουν μηδενικό υπόλοιπο στο λογαριασμό τους.

1. **Tom Brown -> Pin= 1234**
2. **John Mayer -> Pin=1235**
3. **Alisa Brandley -> Pin= 1236**

Από το σημείο αυτό ξεκινά η επεξεργασία της αίτησης του πελάτη.

Κάθε φορά που ένας πελάτης επιτυγχάνει τη σύνδεσή του με το διαχειριστή για να μπει στο σύστημα θα πρέπει να δώσει το κωδικό του (**pin**).

Η εμφάνιση των μηνυμάτων στο τερματικό του πελάτη από τώρα και μέχρι το τέλος του προγράμματος θα γίνονται με τη βοήθεια της συνάρτησης **send()**. Η συνάρτηση αυτή χρησιμοποιείται για την αποστολή μηνυμάτων μέσω ενός socket.

```
msg="Hello! May i have your pin please ?...\n"
```

```
send(newClient_socket,msg,strlen(msg), 0)
```

όπου **newClient_socket** ο file descriptor στον οποίο θα στέλνεται το μήνυμα, **msg** ο δείκτης του πίνακα που περιέχει το μήνυμα και **strlen(msg)** ο αριθμός των bytes που θα σταλούν.

Ο κωδικός του πελάτη αποτελείται από 4 ψηφία και διαβάζεται από το τερματικό του πελάτη με τη μορφή ενός string. Έπειτα αποθηκεύεται σε έναν πίνακα που ονομάζεται

pin. Ο πίνακας αυτός έχει έναν δείκτη ***p**, ο οποίος δείχνει στην διεύθυνση του πρώτου στοιχείου του πίνακα **pin**.

Για να διαβαστεί από το τερματικό του πελάτη ο κωδικός, χρησιμοποιείται η συνάρτηση **recv()**. Η συνάρτηση αυτή είναι υπεύθυνη για την παραλαβή μηνυμάτων μέσω ενός socket.

recv(newClient_socket,p,n+1, 0)

όπου **newClient_socket** ο file descriptor του πελάτη, **p** ο δείκτης του πίνακα στον οποίο θα αποθηκευτεί το μήνυμα που θα ληφθεί και **n+1** μέχρι πόσα bytes μπορούν να διαβαστούν.

Ο κωδικός που θα ληφθεί μπορεί να είναι ενός από τους καταχωρημένους χρήστες, μπορεί να είναι λανθασμένος, δηλαδή να μην υπάρχει στο σύστημα και επίσης μπορεί να είναι και ο τετραψήφιος **9999**. Αυτό σημαίνει ότι, ο χρήστης με τον κωδικό αυτό είναι ο administrator, ένας υπάλληλος της τράπεζας δηλαδή, και το μενού των επιλογών του θα είναι διαφορετικό από αυτό του πελάτη.

Όταν λοιπόν ο κωδικός αποθηκευτεί στον πίνακα **pin** σαν ένα string, πραγματοποιείται έλεγχος χαρακτήρα προς χαρακτήρα με τους αποθηκευμένους σαν string κωδικούς και ανάλογα με τη φύση του κωδικού εμφανίζεται το κατάλληλο μενού ή ζητείτε να πληκτρολογηθεί ξανά από το χρήστη.

Το Μενού των Πελατών

1. Κατάθεση
2. Ανάλυση
3. Ερώτηση Υπολοίπου
4. Ιστορικό Καταθέσεων
5. Ιστορικό Αναλήψεων
6. Έξοδος (Exit)

Το Μενού του administrator

0. Εισαγωγή πελάτη
9. Διαγραφή πελάτη
6. Έξοδος(Exit)

Ο πελάτης καλείται να επιλέξει μία από τις παραπάνω λειτουργίες ανάλογα με το μενού που θα του εμφανιστεί. Η λήψη της απάντησής του, όπως και οποιαδήποτε λήψη από εδώ και πέρα, γίνεται πάλι με την συνάρτηση **recv()** με τη μορφή string,

αποθηκεύοντάς την σε ένα πίνακα με το όνομα **client_request1**, χρησιμοποιώντας το δείκτη ***client_request** του συγκεκριμένου πίνακα.

Πριν από τη λήψη της απάντησης έχει προηγηθεί ο καθαρισμός του πίνακα **pin** με την συνάρτηση **memset(p,0,n+1)**, η οποία θέτει σε αυτόν **n+1** μηδενικά.

c. **Αν ο χρήστης είναι πελάτης :**

1. **Κατάθεση** : Αν ο χρήστης επιλέξει το 1, τότε θα του ζητηθεί να δώσει την σημερινή ημερομηνία κατάθεσης με τη μορφή **xx/yy/zzzz**, η οποία θα αποθηκευτεί αρχικά στον πίνακα **date** χρησιμοποιώντας το δείκτη ***dt** και μετά θα εκχωρηθεί στον δυσδιάστατο πίνακα **pinakas_dates_katathesewn**. Αυτό θα γίνει αφού πρώτα έχει υπολογιστεί η επόμενη ελεύθερη θέση στο πίνακα **pinakas_katathesewn** και στο πίνακα **ypoloipo**.

Έπειτα θα ζητηθεί από το χρήστη να δώσει το ποσό που θέλει να καταθέσει, το οποίο θα διαβαστεί και θα αποθηκευτεί αρχικά σε μορφή string στο πίνακα **use1** μέσω του δείκτη ***use** . Ύστερα με τη βοήθεια της συνάρτησης **atof()**, η οποία μετατρέπει ένα string σε double

ammount=(atof(use))

το ποσό που έδωσε ο χρήστης εκχωρείται στη μεταβλητή **ammount** σε μορφή double πλέον και από αυτήν εκχωρείται στο πίνακα **pinakas_katathesewn**.

Μέσω της μεταβλητής **ammount** αλλάζει κατάλληλα και ο πίνακας **ypoloipo** προσθέτοντας στη νέα θέση το προηγούμενο υπόλοιπο με τη νέα κατάθεση.

Τέλος καθαρίζονται οι πίνακες **date** και **use1** και εμφανίζεται μήνυμα ευχαριστίας. Μετά από αυτό το πρόγραμμα ζητάει από το χρήστη να δώσει ξανά το κωδικό του και να επιλέξει κάποια άλλη λειτουργία του μενού. Αν θέλει να αποσυνδεθεί δίνει απλά την επιλογή 6. Exit και το πρόγραμμα τερματίζεται.

2. **Ανάληψη** : Αν ο χρήστης επιλέξει το 2, θα του ζητηθεί να δώσει το ποσό που θέλει να πάρει από το λογαριασμό του, το οποίο θα διαβαστεί και θα αποθηκευτεί αρχικά σε μορφή string στο πίνακα **use1** μέσω του δείκτη ***use** . Αυτό θα γίνει αφού πρώτα έχει υπολογιστεί η επόμενη ελεύθερη θέση στο πίνακα **pinakas_analypsewn** και στο πίνακα **ypoloipo**.

Ύστερα με τη βοήθεια της συνάρτησης **atof()**, το ποσό που έδωσε ο χρήστης εκχωρείται στη μεταβλητή **ammount** σε μορφή πλέον double.

Έπειτα γίνεται έλεγχος για το αν η αίτηση του ποσού που θέλει να πάρει ο χρήστης μπορεί ή όχι να ικανοποιηθεί. Αν η αίτηση οδηγεί το υπόλοιπό του

λογαριασμού να φτάσει παραπάνω από -3000 τότε δεν μπορεί να ικανοποιηθεί η ανάληψη. Το μείον σημαίνει πως ο χρήστης δεν παραλαμβάνει χρήματα από τις καταθέσεις του, αλλά δανείζεται από την τράπεζα. Κάθε φορά που δανείζεται από την τράπεζα σημαίνει πως το υπόλοιπο του ήταν αρχικά μηδενικό και έπειτα με το δανεισμό πήγε στο μείον. Ο δανεισμός όμως μπορεί να πραγματοποιείται μέχρι ένα συγκεκριμένο ποσό, που στην περίπτωση αυτή είναι 3000. Μόλις τις φτάσει δεν μπορεί να πάρει άλλα χρήματα από την τράπεζα. Έτσι το πρόγραμμα εμφανίζει μήνυμα ευχαριστίας και λέει στο χρήστη να δώσει ξανά το κωδικό του εμφανίζοντας το μενού για την επιλογή μιας άλλης λειτουργίας.

Αν η αίτηση μπορεί να ικανοποιηθεί τότε το ποσό εκχωρείται στο πίνακα **pinakas_analypsewn** διαμορφώνοντας κατάλληλα και τον πίνακα **ypoloipo**, τοποθετώντας στη νέα θέση το προηγούμενο υπόλοιπο μείον το πόσο που ζήτησε ο χρήστης.

Ύστερα ζητείται από το χρήστη να δώσει την σημερινή ημερομηνία ανάληψης με τη μορφή xx/yy/zzzz, η οποία θα αποθηκευτεί αρχικά στον πίνακα **date** χρησιμοποιώντας το δείκτη ***dt** και μετά θα εκχωρηθεί στον δυσδιάστατο πίνακα **pinakas_pinakas_analypsewn**.

Τέλος καθαρίζονται οι πίνακες **date** και **use1** και εμφανίζεται μήνυμα ευχαριστίας. Μετά από αυτό το πρόγραμμα ζητάει από το χρήστη να δώσει ξανά το κωδικό του και να επιλέξει κάποια άλλη λειτουργία του μενού.

3. **Ερώτηση Υπολοίπου** : Αν ο χρήστης επιλέξει το 3, τότε πρώτα υπολογίζεται η τελευταία καταλυμένη θέση στο πίνακα **ypoloipo**. Έπειτα χρησιμοποιείται η συνάρτηση **sprintf()**, η οποία μετατρέπει έναν double σε string

```
sprintf(use2,"%lf",clients[client_position].ypoloipo[thesi_pnks_ypoloipo])
```

όπου **use2** ο πίνακας στον οποίο θα τοποθετηθεί το string, **clients[client_position].ypoloipo[thesi_pnks_ypoloipo]** ο double που θα μετατραπεί, έτσι ώστε να μπορέσει να σταλεί στο τερματικό του πελάτη το

υπόλοιπο σαν ένα μήνυμα με τη βοήθεια της συνάρτησης **send()** όπως προαναφέρθηκε. Έπειτα εμφανίζεται μήνυμα ευχαριστίας και μετά το πρόγραμμα ζητάει από το χρήστη να δώσει ξανά το κωδικό του και να επιλέξει κάποια άλλη λειτουργία του μενού.

4. **Ιστορικό Καταθέσεων** : Αν ο χρήστης επιλέξει το 4, τότε χρησιμοποιώντας έναν βρόγχο επαναλήψεων θα εκτυπωθούν στο τερματικό του πελάτη οι ημερομηνίες καταθέσεων και μετά το σύμβολο « --> » θα ακολουθούν τα ποσά των αντίστοιχων καταθέσεων. Πάλι και σε αυτήν την περίπτωση θα χρησιμοποιηθεί η συνάρτηση **sprintf()**, για την μετατροπή των double ποσών σε string ώστε να μπορέσουν να εκτυπωθούν στη οθόνη. Οι ημερομηνίες από την άλλη μεριά δεν χρειάζονται καμία μετατροπή για να μπορέσουν να εκτυπωθούν.

Τέλος γίνεται εκκαθάριση του πίνακα **use2**. Έπειτα εμφανίζεται μήνυμα ευχαριστίας και μετά το πρόγραμμα ζητάει από το χρήστη να δώσει ξανά το κωδικό του και να επιλέξει κάποια άλλη λειτουργία του μενού.

5. **Ιστορικό Αναλήψεων** : Αν ο χρήστης επιλέξει το 5, τότε χρησιμοποιώντας έναν βρόγχο επαναλήψεων θα εκτυπωθούν στο τερματικό του πελάτη οι ημερομηνίες αναλήψεων και μετά το σύμβολο « --> » θα ακολουθούν τα ποσά των αντίστοιχων αναλήψεων. Πάλι και σε αυτήν την περίπτωση θα χρησιμοποιηθεί η συνάρτηση **sprintf()**, για την μετατροπή των double ποσών σε string ώστε να μπορέσουν να εκτυπωθούν στη οθόνη. Οι ημερομηνίες από την άλλη μεριά δεν χρειάζονται καμία μετατροπή για να μπορέσουν να εκτυπωθούν.

Τέλος γίνεται εκκαθάριση του πίνακα **use2**. Έπειτα εμφανίζεται μήνυμα ευχαριστίας και μετά το πρόγραμμα ζητάει από το χρήστη να δώσει ξανά το κωδικό του και να επιλέξει κάποια άλλη λειτουργία του μενού.

6. **Exit** : Αν ο χρήστης επιλέξει το 6, τότε αποσυνδέεται από το σύστημα και το πρόγραμμα τελειώνει.

d. Αν ο χρήστης είναι ο administrator (pin=9999) :

1. **Εισαγωγή πελάτη και νέου κωδικού** : Αν ο administrator επιλέξει το 0, τότε χρησιμοποιείται η συνάρτηση **realloc()**, η οποία αλλάζει το μέγεθος της μνήμης που είχε οριστεί για το πίνακα **clients**, έτσι ώστε να μπορέσει να προστεθεί ένα ακόμα κελί στο πίνακα αυτόν που θα αντιπροσωπεύει το καινούργιο πελάτη.

```
clients=(struct clients_accounts *)realloc(clients,(number_of_clients+1)*sizeof(struct
clients_accounts))
```

Έπειτα εφόσον έχει δημιουργηθεί η καινούργια δομή, δημιουργείται ξεχωριστά χώρος στη μνήμη και για τα περιεχόμενα της δομής αυτής. Στη συνέχεια μέσω της χρήσης των συναρτήσεων **recv()** και **send()** συμπληρώνονται τα στοιχεία του νέου πελάτη, δηλαδή το όνομα (**first_name**), το επώνυμο (**last_name**) και ο κωδικός του (**account_number**). Μετά από την εισαγωγή του ονόματος, του επωνύμου και του κωδικού γίνεται και εκκαθάριση των αντίστοιχων πινάκων. Ακολουθεί επίσης και η αρχικοποίηση των υπολοίπων πινάκων της δομής του νέου πελάτη, δηλαδή **ypoloipo**, **pinakas_katathesewn** κτλ. Τέλος εμφανίζεται μήνυμα ευχαριστίας και μετά το πρόγραμμα ζητάει από το χρήστη να δώσει ξανά το κωδικό του και να επιλέξει κάποια άλλη λειτουργία του μενού.

9. **Διαγραφή πελάτη** : Αν ο administrator επιλέξει το 9, τότε ζητείται από το χρήστη να πληκτρολογήσει το κωδικό του πελάτη που θα ήθελε να διαγράψει.

Έτσι λοιπόν μόλις ο πελάτης αυτός εντοπιστεί μέσα από έναν βρόγχο επαναλήψεων, χρησιμοποιείται η συνάρτηση **free()**, η οποία θα απελευθερώσει τη μνήμη που έχει δεσμευτεί για τα στοιχεία του πελάτη αυτού.

```
free(&(clients[i]))
```

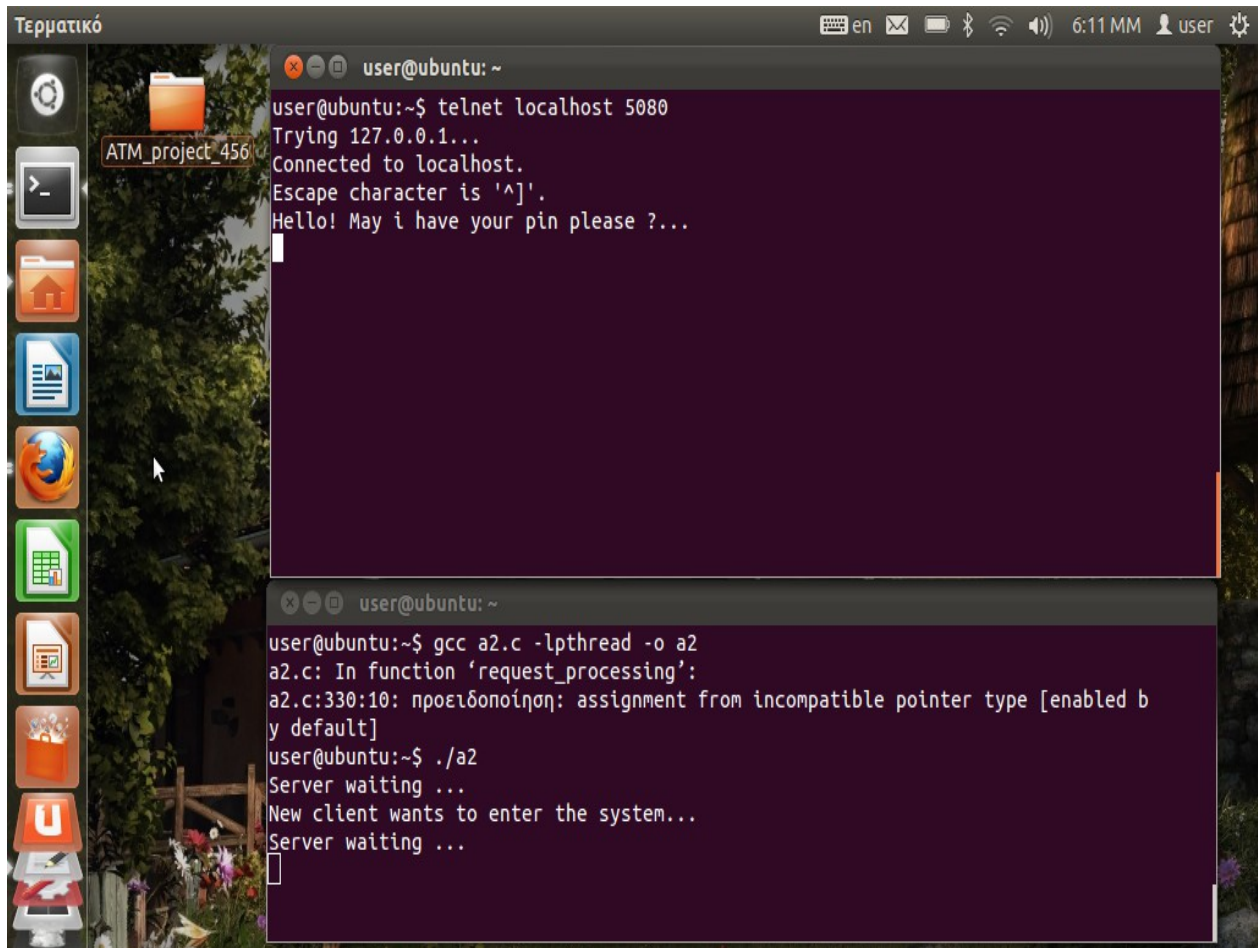
Έπειτα εμφανίζεται μήνυμα επιβεβαίωσης της διαγραφής και γίνεται εκκαθάριση του πίνακα που περιείχε το κωδικό του πελάτη που διαγράφηκε, δηλαδή του **new_pin**.

Τέλος εμφανίζεται μήνυμα ευχαριστίας και μετά το πρόγραμμα ζητάει από το χρήστη να δώσει ξανά το κωδικό του και να επιλέξει κάποια άλλη λειτουργία του μενού.

6. **Exit** : Αν ο χρήστης επιλέξει το 6, τότε αποσυνδέεται από το σύστημα και το πρόγραμμα τελειώνει.

4. Screenshots

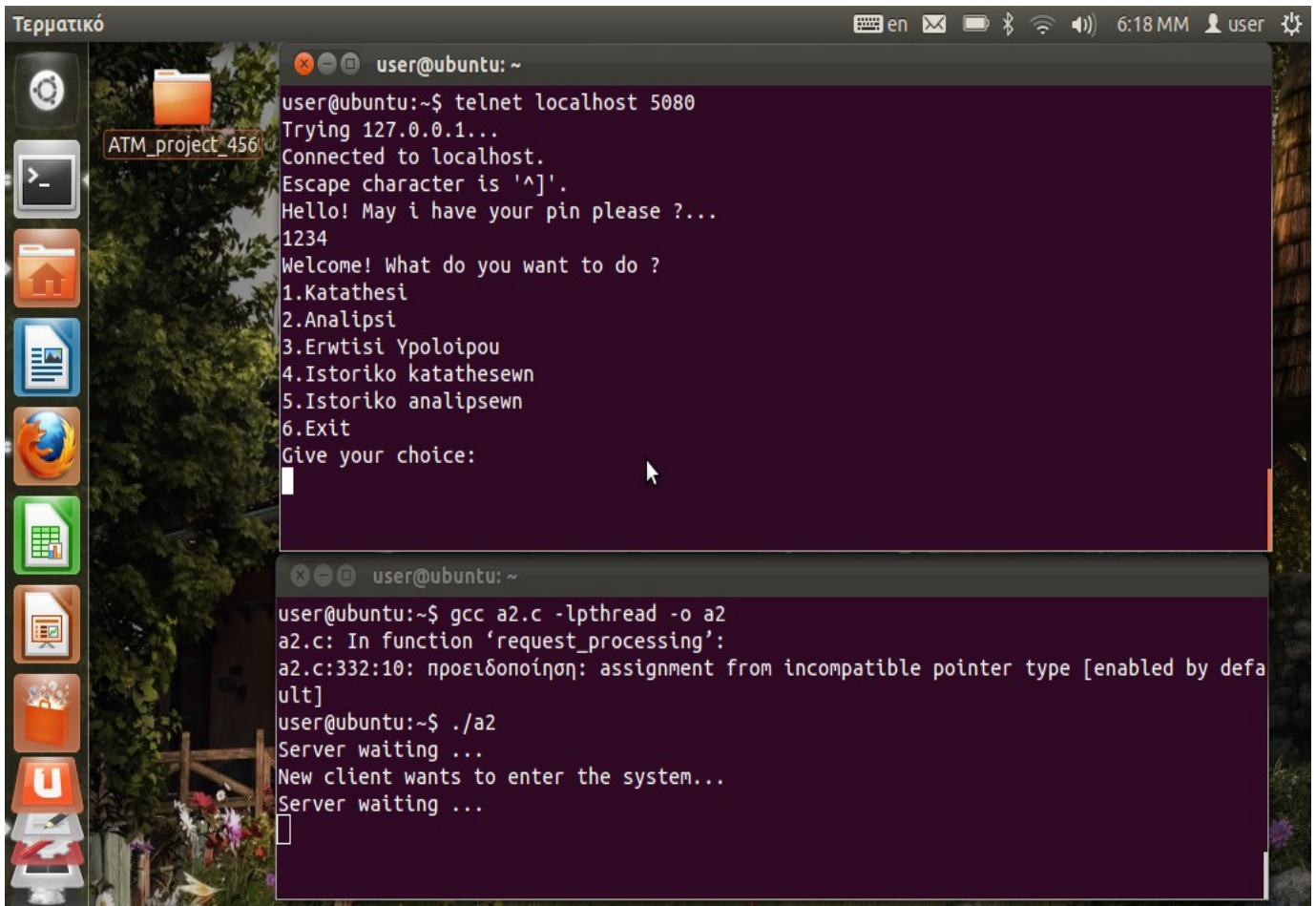
Το πρώτο τερματικό αντιπροσωπεύει το πελάτη (client) και το δεύτερο το διαχειριστή (server) της τράπεζας. Εδώ έγινε μια επιτυχής σύνδεση.



```
user@ubuntu:~$ telnet localhost 5080
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Hello! May i have your pin please ?...
^_

user@ubuntu:~$ gcc a2.c -lpthread -o a2
a2.c: In function 'request_processing':
a2.c:330:10: προειδοποίηση: assignment from incompatible pointer type [enabled by default]
user@ubuntu:~$ ./a2
Server waiting ...
New client wants to enter the system...
Server waiting ...
^_
```

Εδώ συνδέθηκε ο καταχωρημένος χρήστης Tom Brown.



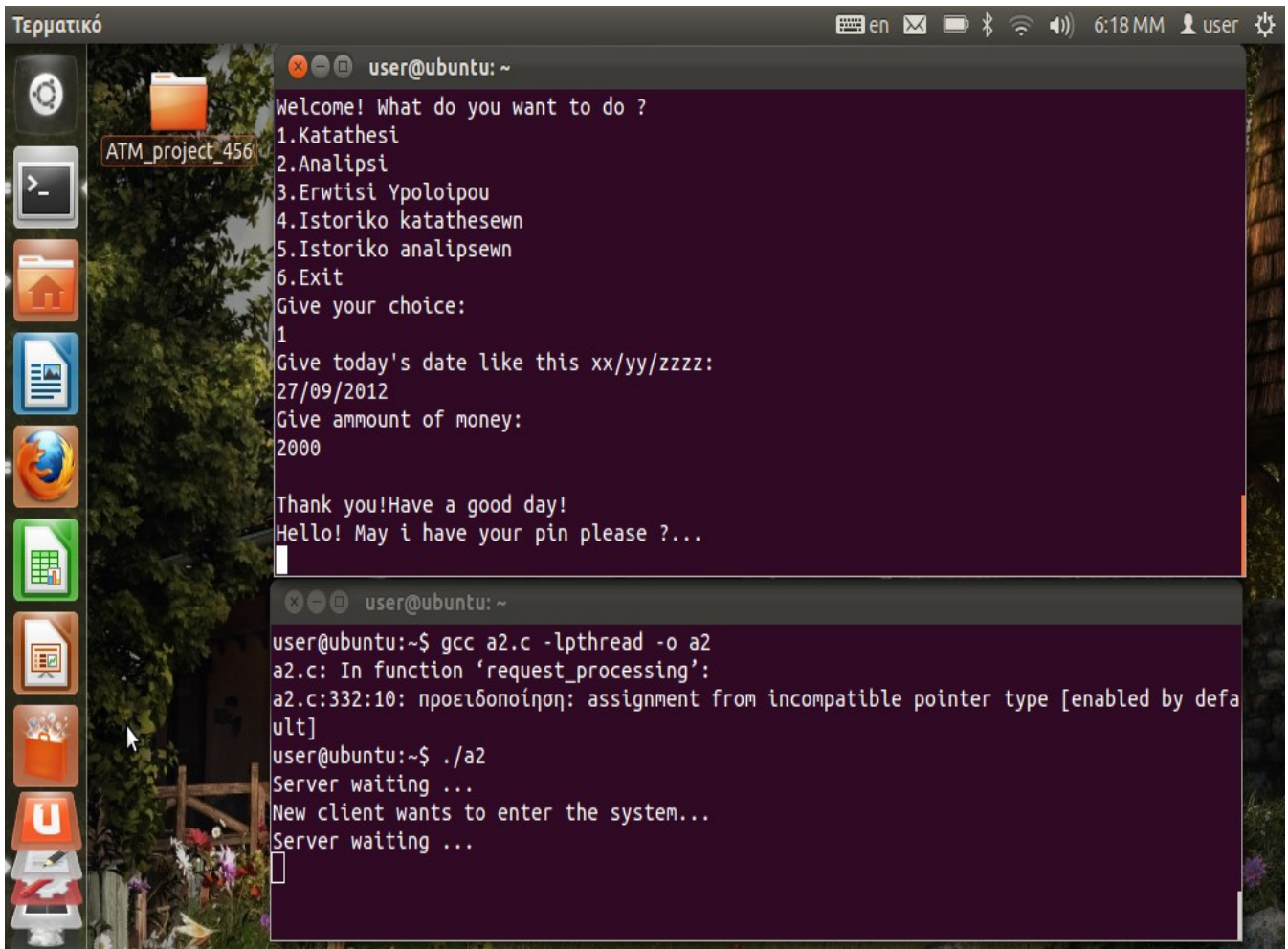
The screenshot shows an Ubuntu desktop environment with a terminal window open. The terminal displays the following output:

```
user@ubuntu:~$ telnet localhost 5080
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Hello! May i have your pin please ?...
1234
Welcome! What do you want to do ?
1.Katathesi
2.Analipsi
3.Erwtsi Ypoloipou
4.Istoriko katathesewn
5.Istoriko analipsewn
6.Exit
Give your choice:
█
```

The terminal then shows the compilation and execution of a C program:

```
user@ubuntu:~$ gcc a2.c -lpthread -o a2
a2.c: In function 'request_processing':
a2.c:332:10: προειδοποίηση: assignment from incompatible pointer type [enabled by default]
user@ubuntu:~$ ./a2
Server waiting ...
New client wants to enter the system...
Server waiting ...
█
```

Εδώ ο Tom Brown επέλεξε να κάνει μια κατάθεση των 2000 ευρώ.



The screenshot shows an Ubuntu desktop environment with the Greek word "Τερατικό" (Terrible) in the top-left corner. The top panel displays system icons for keyboard layout (en), mail, battery, Bluetooth, Wi-Fi, and volume, along with the time "6:18 MM" and the user "user". The desktop background is a scenic image of a garden. A vertical dock on the left contains icons for Dash, Home, Files, Firefox, LibreOffice Calc, LibreOffice Impress, LibreOffice Writer, and the Dash home icon. A terminal window titled "user@ubuntu: ~" is open, displaying the following text:

```
user@ubuntu: ~  
Welcome! What do you want to do ?  
1.Katathesi  
2.Analipsi  
3.Erwtsisi Ypolojiou  
4.Istoriko katathesewn  
5.Istoriko analipsewn  
6.Exit  
Give your choice:  
1  
Give today's date like this xx/yy/yyyy:  
27/09/2012  
Give ammount of money:  
2000  
Thank you!Have a good day!  
Hello! May i have your pin please ?...  
  
user@ubuntu:~$ gcc a2.c -lpthread -o a2  
a2.c: In function 'request_processing':  
a2.c:332:10: προειδοποίηση: assignment from incompatible pointer type [enabled by default]  
user@ubuntu:~$ ./a2  
Server waiting ...  
New client wants to enter the system...  
Server waiting ...  
█
```

Εδώ ο Tom Brown στα δεξιά , επιλέγει να δει το υπόλοιπο του λογαριασμού του μετά την κατάθεση και στα αριστερά αυτό εμφανίζεται.

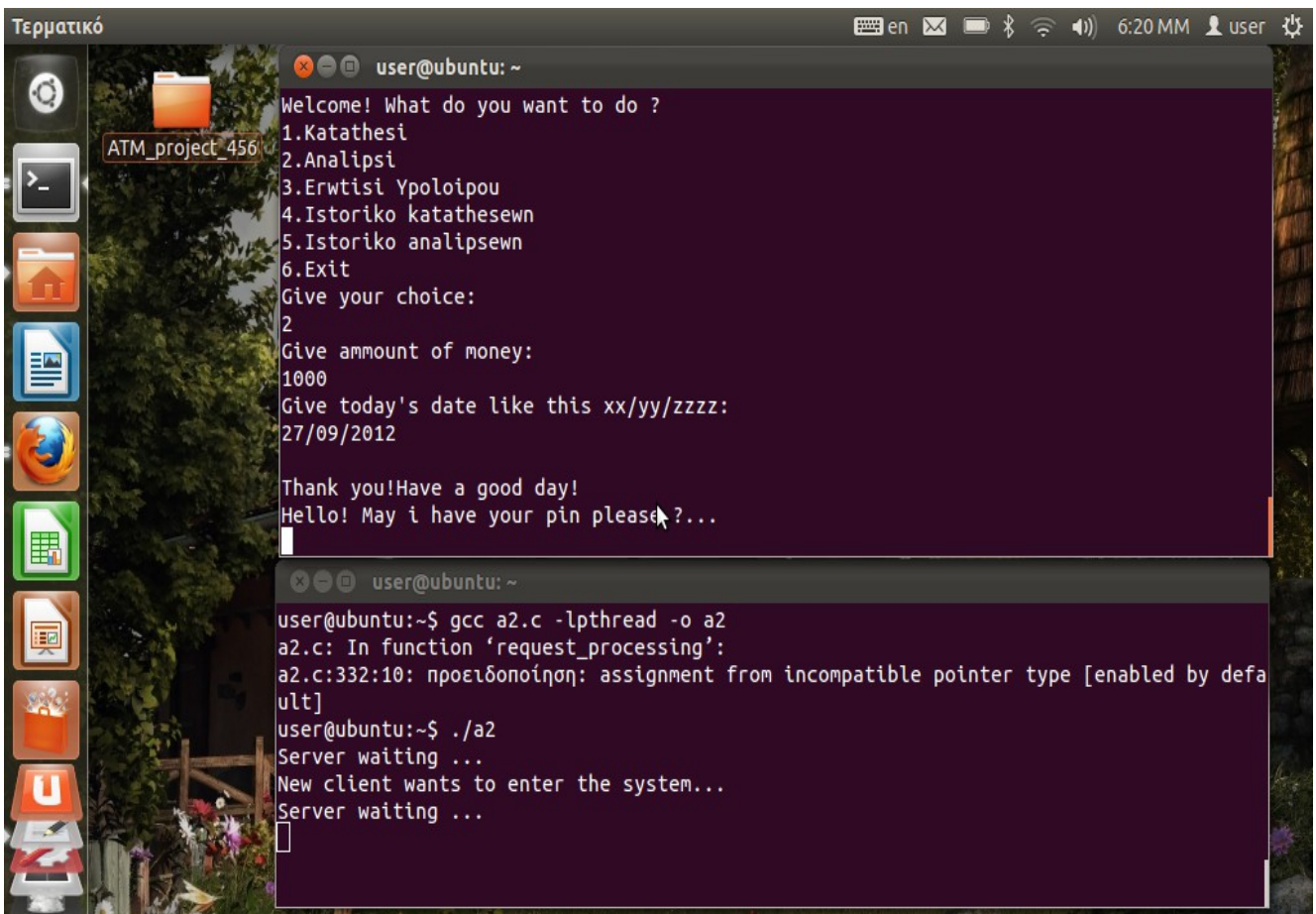
```

user@ubuntu:~$ gcc a2.c -lthread -o a2
a2.c: In function 'request_processing':
a2.c:332:10: προειδοποίηση: assignment from incompatible pointer type [enabled by default]
user@ubuntu:~$ ./a2
Server waiting ...
New client wants to enter the system...
Server waiting ...

user@ubuntu:~$ gcc a2.c -lthread -o a2
a2.c: In function 'request_processing':
a2.c:332:10: προειδοποίηση: assignment from incompatible pointer type [enabled by default]
user@ubuntu:~$ ./a2
Server waiting ...
New client wants to enter the system...
Server waiting ...

```

Εδώ ο Tom Brown επιλέγει να κάνει ανάληψη του ποσού των 1000 ευρώ αυτή τη φορά.



The screenshot shows an Ubuntu desktop with a terminal window open. The terminal displays a menu-driven program in Greek. The user selects option 2, enters 1000 for the amount, and 27/09/2012 for the date. The program then asks for a PIN. Below this, the user runs a compiler command, which results in an error: 'assignment from incompatible pointer type'. Finally, the user runs the executable, which outputs 'Server waiting ...' and 'New client wants to enter the system...'.

```
user@ubuntu: ~  
Welcome! What do you want to do ?  
1.Katathesi  
2.Analipsi  
3.Erwtsi Ypoloipou  
4.Istoriko katathesewn  
5.Istoriko analipsewn  
6.Exit  
Give your choice:  
2  
Give ammount of money:  
1000  
Give today's date like this xx/yy/yyyy:  
27/09/2012  
Thank you!Have a good day!  
Hello! May i have your pin please?...  
  
user@ubuntu:~$ gcc a2.c -lpthread -o a2  
a2.c: In function 'request_processing':  
a2.c:332:10: προειδοποίηση: assignment from incompatible pointer type [enabled by default]  
user@ubuntu:~$ ./a2  
Server waiting ...  
New client wants to enter the system...  
Server waiting ...  
█
```

Εδώ ο Tom Brown επιλέγει να δει ξανά το υπόλοιπο του μετά την κατάθεση. Στα δεξιά φαίνεται η επιλογή του και στα αριστερά εμφανίζεται το υπόλοιπο.

```

user@ubuntu:~$ gcc a2.c -lthread -o a2
a2.c: In function 'request_processing':
a2.c:332:10: note: 'assignment from incompatible pointer type [enabled by default]'
user@ubuntu:~$ ./a2
Server waiting ...
New client wants to enter the system...
Server waiting ...

Give amount of money:
1000
Give today's date like this xx/yy/zzzz:
27/09/2012

Thank you!Have a good day!
Hello! May i have your pin please ?...
1234
Welcome! What do you want to do ?
1.Katathesi
2.Analipsi
3.Erwitisi Ypotoipou
4.Istoriko katathesewn
5.Istoriko analipsewn
6.Exit
Give your choice:
3

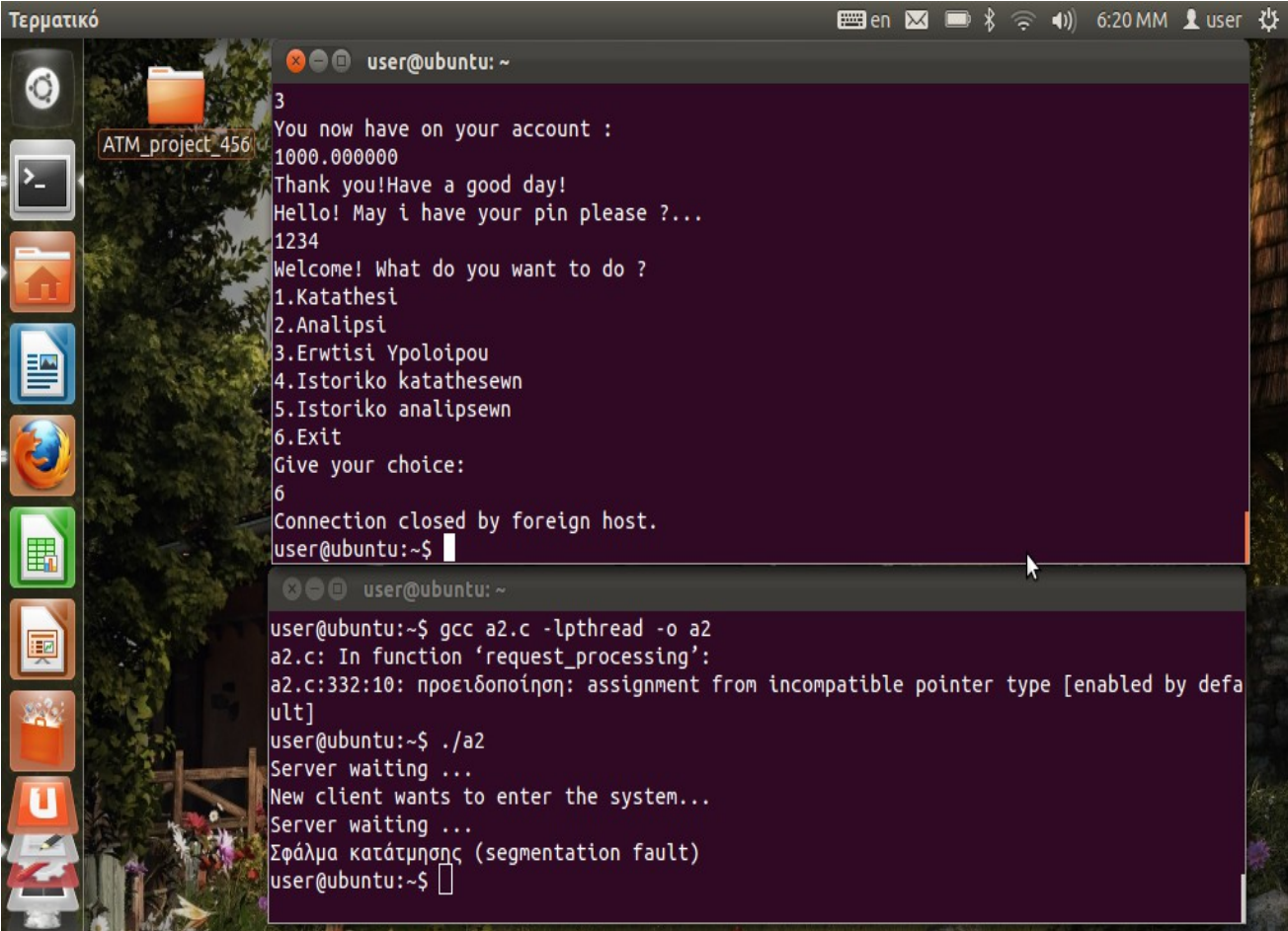
user@ubuntu:~$ gcc a2.c -lthread -o a2
a2.c: In function 'request_processing':
a2.c:332:10: note: 'assignment from incompatible pointer type [enabled by default]'
user@ubuntu:~$ ./a2
Server waiting ...
New client wants to enter the system...
Server waiting ...

Thank you!Have a good day!
Hello! May i have your pin please ?...
1234
Welcome! What do you want to do ?
1.Katathesi
2.Analipsi
3.Erwitisi Ypotoipou
4.Istoriko katathesewn
5.Istoriko analipsewn
6.Exit
Give your choice:
3

You now have on your account :
1000.000000
Thank you!Have a good day!
Hello! May i have your pin please ?...

```

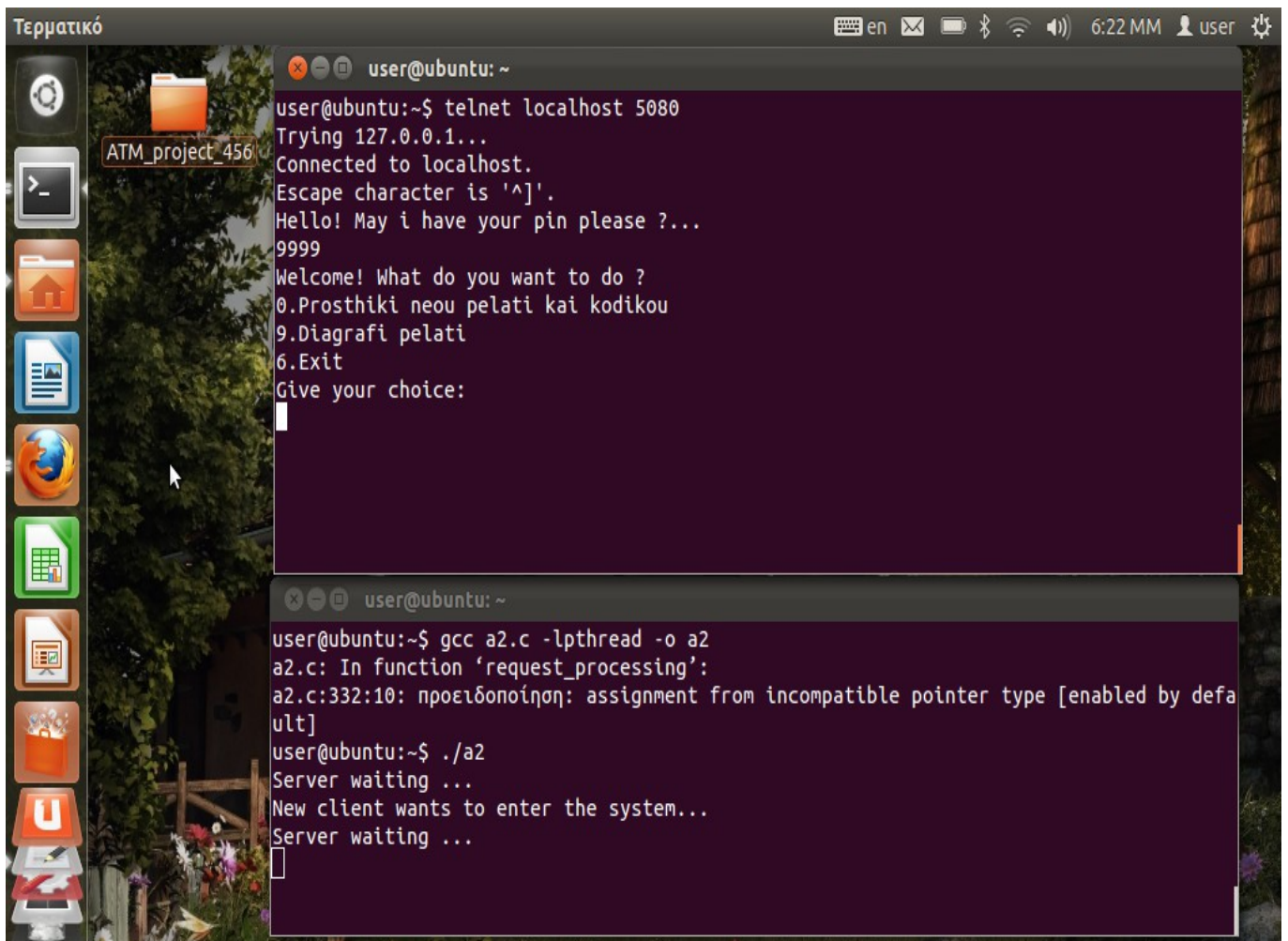
Ο Tom Brown επιλέγει την αποσύνδεσή του από το σύστημα.



```
Τερματικό en 6:20 MM user
user@ubuntu: ~
3
You now have on your account :
1000.000000
Thank you!Have a good day!
Hello! May i have your pin please ?...
1234
Welcome! What do you want to do ?
1.Katathesi
2.Analipsi
3.Erwtsi Ypoloipou
4.Istoriko katathesewn
5.Istoriko analipsewn
6.Exit
Give your choice:
6
Connection closed by foreign host.
user@ubuntu:~$

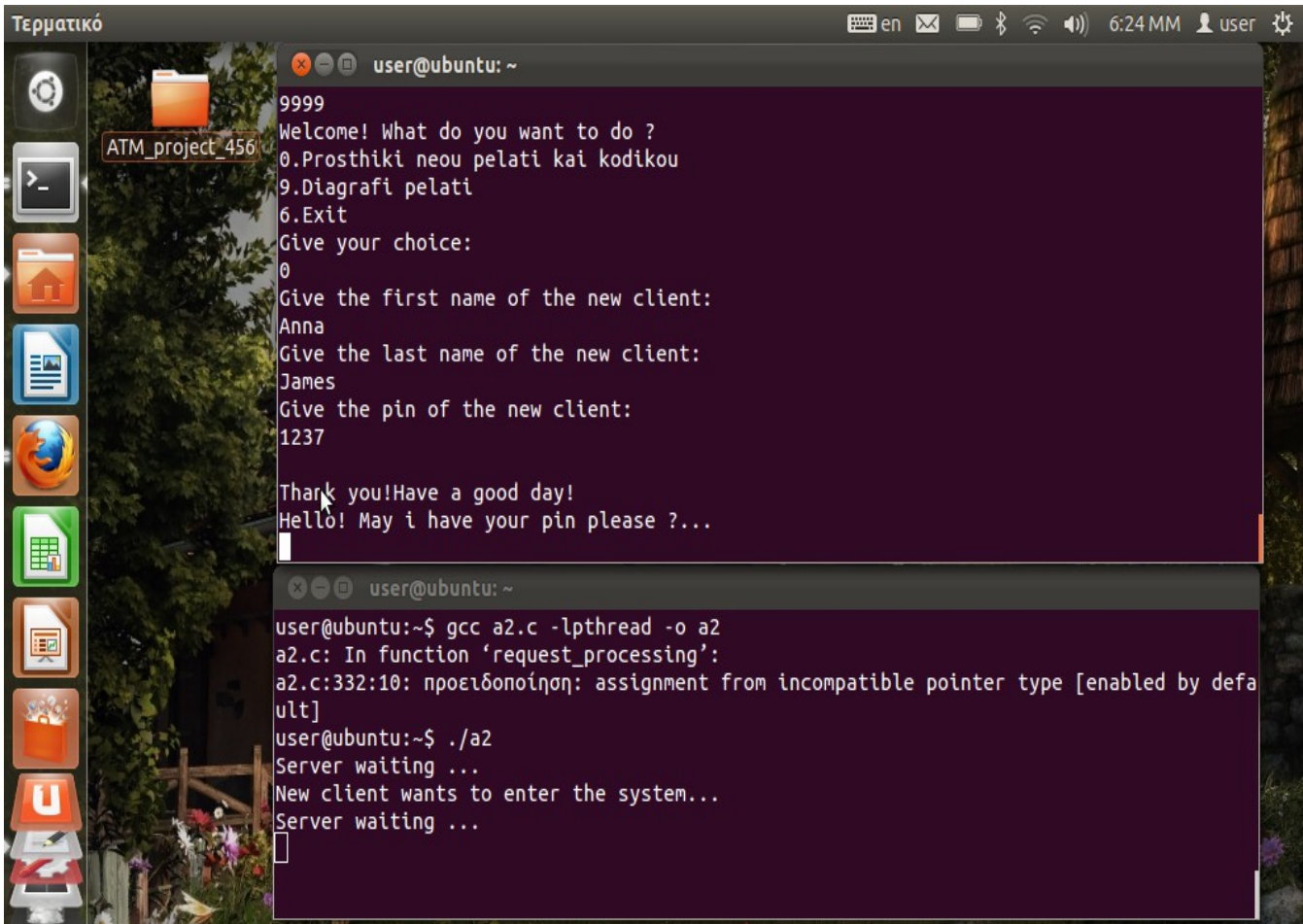
user@ubuntu:~$ gcc a2.c -lpthread -o a2
a2.c: In function 'request_processing':
a2.c:332:10: προειδοποίηση: assignment from incompatible pointer type [enabled by default]
user@ubuntu:~$ ./a2
Server waiting ...
New client wants to enter the system...
Server waiting ...
Σφάλμα κατάτμησης (segmentation fault)
user@ubuntu:~$
```

Εδώ φαίνεται η σύνδεση του administrator με κωδικό 9999.



```
user@ubuntu: ~  
user@ubuntu:~$ telnet localhost 5080  
Trying 127.0.0.1...  
Connected to localhost.  
Escape character is '^]'.  
Hello! May i have your pin please ?...  
9999  
Welcome! What do you want to do ?  
0.Prosthiki neou pelati kai kodikou  
9.Diagrafi pelati  
6.Exit  
Give your choice:  
█  
  
user@ubuntu:~$ gcc a2.c -lpthread -o a2  
a2.c: In function 'request_processing':  
a2.c:332:10: προειδοποίηση: assignment from incompatible pointer type [enabled by default]  
user@ubuntu:~$ ./a2  
Server waiting ...  
New client wants to enter the system...  
Server waiting ...  
█
```

Εδώ ο administrator επιλέγει να εισάγει έναν νέο πελάτη με το όνομα Anna James και κωδικό 1237.



The screenshot shows a terminal window titled "user@ubuntu: ~" with a dark purple background. The terminal output is as follows:

```
9999
Welcome! What do you want to do ?
0.Prosthiki neou pelati kai kodikou
9.Diagrafi pelati
6.Exit
Give your choice:
0
Give the first name of the new client:
Anna
Give the last name of the new client:
James
Give the pin of the new client:
1237

Thank you!Have a good day!
Hello! May i have your pin please ?...

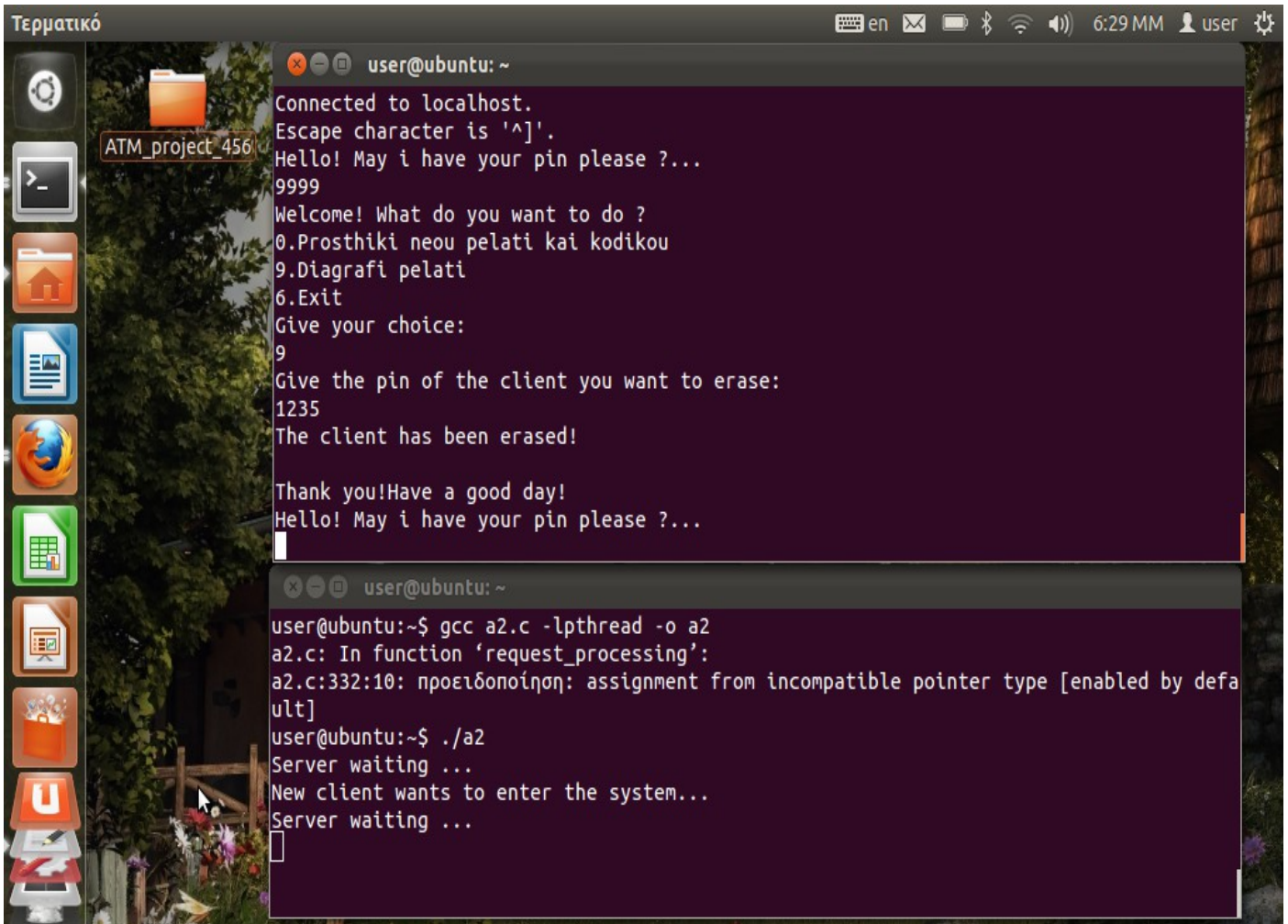
```

Below this, a second terminal window shows the compilation and execution of a program:

```
user@ubuntu:~$ gcc a2.c -lpthread -o a2
a2.c: In function 'request_processing':
a2.c:332:10: προειδοποίηση: assignment from incompatible pointer type [enabled by default]
user@ubuntu:~$ ./a2
Server waiting ...
New client wants to enter the system...
Server waiting ...

```


Εδώ ο administrator επιλέγει να διαγράψει από το σύστημα το πελάτη με κωδικό 1235, δηλαδή τον John Mayer.



```
user@ubuntu: ~  
Connected to localhost.  
Escape character is '^]'.  
Hello! May i have your pin please ?...  
9999  
Welcome! What do you want to do ?  
0.Prosthiki neou pelati kai kodikou  
9.Diagrafi pelati  
6.Exit  
Give your choice:  
9  
Give the pin of the client you want to erase:  
1235  
The client has been erased!  
Thank you!Have a good day!  
Hello! May i have your pin please ?...  
  
user@ubuntu:~$ gcc a2.c -lpthread -o a2  
a2.c: In function 'request_processing':  
a2.c:332:10: προειδοποίηση: assignment from incompatible pointer type [enabled by default]  
user@ubuntu:~$ ./a2  
Server waiting ...  
New client wants to enter the system...  
Server waiting ...
```

5. Σημειώσεις

2. Τα σχόλια στο κώδικα του προγράμματος μου είναι γραμμένα στα ελληνικά με λατινικούς χαρακτήρες και όχι με ελληνικούς, καθώς δεν ήξερα εάν θα μπορούσαν να διαβαστούν σωστά σε διαφορετικό υπολογιστή εξαιτίας του compiler. Για το λόγο αυτό με συγχωρείται για τη δυσκολία της ανάγνωσής τους.