

Λειτουργικά Συστήματα Ι

Τομέας Λογικού των Υπολογιστών

Δούκας Ιωάννης
5509
douka@ceid.upatras.gr

Νικολάου Γλαύκος
5693
nikolaog@ceid.upatras.gr

1 Στόχος της Άσκησης

Στόχος της άσκησης είναι η εξοικείωση με θεμελιώδεις έννοιες και μηχανισμούς που προσφέρει το λειτουργικό σύστημα. Για το σκοπό αυτό κάνουμε δύο διαφορετικές υλοποιήσεις. Η μία βασίζεται στη δημιουργία και διαχείριση διεργασιών (processes) και την επικοινωνία /συγχρονισμό μεταξύ τους με IPC (Interprocess Communication), όπως shared memory, semaphores, signals κ.τ.λ. Η άλλη υλοποίηση γίνεται με τη χρήση νημάτων (threads), και το συγχρονισμό μεταξύ τους για την σωστή διαχείριση της εκ των πραγμάτων κοινής τους μνήμης με mutexes και condition variables. Και οι δύο υλοποιήσεις χρησιμοποιούν sockets για την επικοινωνία client-server

2 Γενική Περιγραφή

Στην παρούσα εργασία υλοποιήσαμε έναν server για ένα multiplayer online game με συγκεκριμένο αριθμό παιχτών. Κάθε καινούριος παίχτης που συνδέεται επιλέγει στοιχεία για το αρχικό του inventory. Μόλις συμπληρωθεί ο προκαθορισμένος αριθμός παιχτών ξεκινάει μία καινούρια παρτίδα η οποία θα εκτελεστεί παράλληλα με τις προηγούμενες είτε μέσω διεργασίας (fork) ή μέσω νημάτων (threads). Για κάθε παρτίδα υπάρχει ένα socket μέσω του οποίου μπορούν οι παίχτες να επικοινωνούν μεταξύ τους και να ανταλλάσσουν μηνύματα. Η παρτίδα τελειώνει μόλις και ο τελευταίος παίχτης (της παρτίδας) αποχωρίσει από το παιχνίδι.

3 Περιγραφή Server-Client μοντέλου

Και στις δύο υλοποιήσεις, με διεργασίες και με νήματα, έχουμε χρησιμοποιήσει το μοντέλο server-client.

3.1 Server

Ο server εκτελείται ως εξής: `./gameserver -p <num_of_players> -i <game_inventory> -q <quota_per_player>` Η πρώτη παράμετρος είναι ένας integer που δηλώνει τον αριθμό παικτών ανά παρτίδα και η δεύτερη παρέχει το filename με το συνολικό απόθεμα (inventory) ανά παρτίδα. Η τρίτη παράμετρος (quota) καθορίζει το μέγιστο μέγεθος inventory που αναλογεί σε κάθε παίχτη. Και οι τρεις παράμετροι παραμένουν ίδιες για όλες τις παρτίδες που θα ξεκινήσει αυτό το instance του server. Για κάθε παρτίδα το inventory είναι αναλώσιμο. Για να δοθούν κάποια αντικείμενα σε κάποιο παίχτη πρέπει να είναι διαθέσιμα. Στην περίπτωση που κάποιο από τα αντικείμενα τα οποία ζητάει ο παίκτης είναι άγνωστο στον server ή αν έχει ήδη εξαντληθεί, ή αν κάποιος παίκτης ζητάει περισσότερα αντικείμενα από το quota του ο server θα του επιστρέψει ένα μήνυμα λάθους και θα διακόψει άμεσα τη σύνδεση με τον συγκεκριμένο παίκτη. Μέχρι να συγκεντρωθεί ο απαιτούμενος αριθμός παικτών για να ξεκινήσει μία παρτίδα, οι ήδη συνδεδεμένοι παίκτες λαμβάνουν ένα μήνυμα αναμονής κάθε 5 sec. Για κάθε καινούρια παρτίδα το συνολικό inventory του server αρχικοποιείται στις αρχικές τιμές. Η έναρξη της παρτίδας στους παίκτες σηματοδοτείτε λαμβάνοντας από τον server το μήνυμα "START". Κατά την διάρκεια μίας παρτίδας ότι μήνυμα στέλνεται από κάποιον παίκτη στον server, προωθείται από τον server προς όλους τους υπόλοιπους παίκτες της ίδιας παρτίδας. Ο gameserver τερματίζεται με Ctrl-C.

3.2 Client

Στη συγκεκριμένη περίπτωση οι clients οι οποίοι θα συνδεθούν στον server είναι οι παίκτες, οι οποίοι εκτελούνται ως εξής: `./player -n <name> -i <inventory> <server_host>` Η πρώτη παράμετρος δηλώνει το όνομα του παίχτη, η δεύτερη το inventory που επιλέγει ο παίχτης για τον εαυτό του, και η τρίτη το hostname όπου τρέχει ο gameserver. Κάθε παίχτης δικαιούται να ζητήσει μέχρι quota αντικείμενα για το δικό του inventory.

3.3 Επικοινωνία Server-Client

Η επικοινωνία μεταξύ παίκτη και server γίνεται με TPC, και το πρωτόκολλο επικοινωνίας είναι σε ASCII. Συγκεκριμένα ο, ο παίκτης ανοίγει μία σύνδεση TCP με τον server, και στέλνει μερικές γραμμές με ASCII δεδομένα, μέσω ενός socket. Η πρώτη γραμμή περιέχει απλά το όνομα του, και οι επόμενες τα αντικείμενα που επιλέγει. Κατόπιν, στέλνει μία κενή γραμμή, και αυτό σηματοδοτεί το τέλος της αίτησης συμμετοχής του. Ο server απαντά με "OK" ή με σχετικό μήνυμα λάθους, ακολουθούμενο από \n.

4 Αναλυτικές Υλοποίηση με βάση διεργασίες

4.1 Περιγραφή Υλοποίησης

Για την υλοποίηση με βάση τις διεργασίες χρησιμοποιήσαμε τις κλήσεις συστήματος fork, shmget, και sem_open, για τη δημιουργία νέων διεργασιών, κοινής μνήμης και σηματοφόρους αντίστοιχα. Αρχικά ο server ελέγχει τις παραμέτρους που δόθηκαν από τη γραμμή εντολών, αρχικοποιεί τις κοινές μνήμες και τους σηματοφόρους και στη συνέχεια εμφανίζει μήνυμα ότι είναι έτοιμος να δεχτεί παίκτες. Η αποδοχή των clients γίνεται μέσα σε έναν ατέρμων βρόχο.

Κάθε τραπέζι και κάθε νέα σύνδεση με παίκτη είναι μία καινούρια διεργασία για τον server. Κατά την σύνδεση του πρώτου παίκτη ανοίγει μία καινούρια διεργασία η οποία είναι η διεργασία για το τραπέζι 1. Επίσης ανοίγει και διεργασία η οποία θα αναλάβει την επικοινωνία με τον client. "Accept()" μέσα στον server μπορεί να γίνει σε δύο σημεία. Το πρώτο είναι πριν ανοίξει καινούριο τραπέζι, και το άλλο είναι κάτω από την διεργασία τραπέζι για την οποία διεκδικούν μία θέση οι clients.

Στον server έχουμε σε κοινή μνήμη, μεταξύ των διεργασιών του, το inventory της τρέχουσας παρτίδας, τον συνολικό αριθμό παιχτών που έχουν γίνει αποδεκτοί και ένα πίνακα για κάθε τραπέζι με τα socket descripts των clients που είναι συνδεδεμένοι, ένα μετρητή που κρατάει πόσοι είναι οι παίκτες μέσα στο τραπέζι και ένα string buffer. Το πίνακα με τα socket descriptors τον χρειαζόμαστε έτσι ώστε κάθε φορά που στέλνει μήνυμα κάποιος client, η διεργασία αυτού στον server στέλνει ένα σήμα SIGUSR1 το οποίο ενεργοποιεί μία συνάρτηση που στέλνει το μήνυμα αυτό, το οποίο έχει αποθηκευτεί στον buffer, στους υπόλοιπους παίκτες. Τους δύο μετρητές τους χρησιμοποιούμε για να γνωρίζουμε πότε πρέπει να ξεκινήσει νέα παρτίδα. Κάθε κοινή μνήμη συνοδεύεται και από το σηματοφόρο της για την σωστή διαχείριση της κρίσιμης περιοχής που δημιουργείται κάθε φορά.

4.2 Παράδειγμα εκτέλεσης

Παρακάτω παρουσιάζεται παράδειγμα εκτέλεσης. Στο συγκεκριμένο παράδειγμα έχουμε ανοίξει τον server: `./server -p 2 -i inv_server.txt -q 10` και 4 με ονόματα giannis, glafkos, spyros και andreas.

```
./servrer -p 2 testing/inv_server.txt -q 10
```

```
For exit press Ctr-C
```

```
Starting inventory:
```

```
gold    10
```

```
armor   50
```

```
ammo    40
```

```
lumber  20
```

```
magic   15
```

```
rock    50
```

```
Quota per player: 10
```

```
Players per table: 2
```

```
+----- NEW TABLE: 2502 -----+
```

```
Message from client (2503):
```

```
giannis
```

```
gold    1
```

```
rock    1
```

```
1 players in 2502 table
```

```
New inventory for 2502 table
```

```
gold    9
```

```
armor   50
```

```
ammo    40
```

```
lumber  20
```

```
magic   15
```

```
rock    49
```

```
Message from client (2553):
```

```
glafkos
```

```
gold    1
```

```
rock    1
```

2 players in 2502 table

New inventory for 2502 table

gold	8
armor	50
ammo	40
lumber	20
magic	15
rock	48

+----- NEW TABLE: 2606 -----+

Message from client (2607):

spyros

gold	1
rock	1

1 players in 2606 table

New inventory for 2606 table

gold	9
armor	50
ammo	40
lumber	20
magic	15
rock	47

Message from client (2658):

andreas

gold	1
rock	1

2 players in 2606 table

New inventory for 2606 table

gold	8
armor	50
ammo	40
lumber	20
magic	15
rock	46

- giannis (2503) from 2502 said: hello

- giannis (2503) from 2502 said: my name is giannis
- glafkos (2553) from 2502 said: hello
- glafkos (2553) from 2502 said: my name is glafkos
- spyros (2607) from 2606 said: hello
- andreas (2658) from 2606 said: hello
- spyros (2607) from 2606 said: my name is spyros
- andreas (2658) from 2606 said: my name is andreas
- giannis (2503) from 2502 said: bye

Client giannis (2503) left from the table

- glafkos (2553) from 2502 said: bye

Client glafkos (2553) left from the table
Table 2502 closed

- spyros (2607) from 2606 said: bye

Client spyros (2607) left from the table

- andreas (2658) from 2606 said: bye

Client andreas (2658) left from the table
Table 2606 closed

^CSigno: 2

BYE

Signo: 2

BYE

Child 2606 terminated. Signo 17

```
./client -n giannis -i testing/inv_client.txt localhost
```

```
giannis
```

```
gold      1
```

```
rock      1
```

```
Server 's answer : OK
```

```
Server 's answer : please wait
```

```
Server 's answer : please wait
```

```
Server 's answer : please wait
```

```
Server 's answer : please wait
```

```
Server 's answer : START
```

```
hello
```

```
- giannis: hello
```

```
my name is giannis
```

```
- giannis: my name is giannis
```

```
- glafkos: hello
```

```
- glafkos: my name is glafkos
```

```
bye
```

```
- giannis: bye
```

```
^C
```

```
./client -n glafkos -i testing/inv_client.txt localhost
```

```
glafkos
```

```
gold      1
```

```
rock      1
```

```
Server 's answer : OK
```

```
Server 's answer : START
```

```
- giannis: hello
```

```
- giannis: my name is giannis
```

```
hello
```

```
- glafkos: hello
```

```
my name is glafkos
```

```
- glafkos: my name is glafkos
```

```
- giannis: bye
```

```
- Message from server: giannis has left from the table  
bye
```

```
- glafkos: bye
```

```
^C
```

```
./client -n andreas -i testing/inv_client.txt localhost
```

```
andreas
```

```
gold      1
```

```
rock      1
```

```
Server 's answer : OK
```



```
Server 's answer : START
- spyros: hello

hello
- andreas: hello

- spyros: my name is spyros
my name is andreas
- andreas: my name is andreas

- spyros: bye

- Message from server: spyros has left from the table
bye
- andreas: bye

^C
```

```
./client -n spyros -i testing/inv_client.txt localhost
```

```
spyros
gold      1
rock      1

Server 's answer : OK
Server 's answer : please wait
Server 's answer : please wait
Server 's answer : please wait
Server 's answer : please wait
Server 's answer : START
hello
- spyros: hello

- andreas: hello
```

```
my name is spyros
```

```
- spyros: my name is spyros
```

```
- andreas: my name is andreas
```

```
bye
```

```
- spyros: bye
```

```
^C
```