



**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ
ΣΥΣΤΗΜΑΤΩΝ**

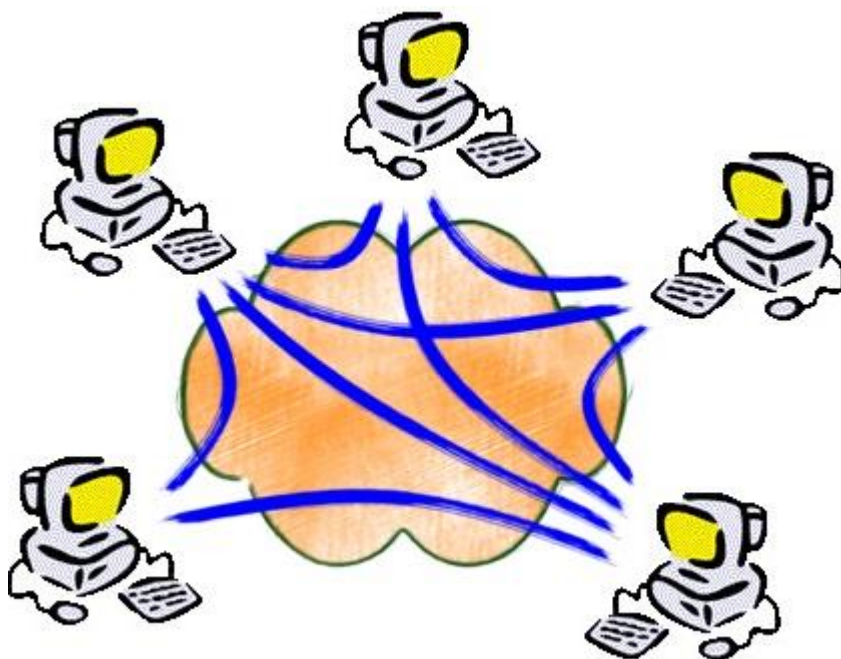
Μάθημα: Κατανεμημένα Συστήματα

Εργαστηριακός Διδάσκων: Αναγνωστόπουλος Μ. – Δαμόπουλος Δ.

3^η ομαδική Εργασία

Μέλοι ομάδας εργασίας:

Μαρία Θεοδωράκη Α.Μ.:321/2008041
Βερροιοπούλου Αθηνά Α.Μ.:321/2008011



Ημερομηνία Παράδοσης : 25/05/2013

ΖΗΤΟΥΜΕΝΑ

A)Υλοποίηση

Απάντηση(a – f):

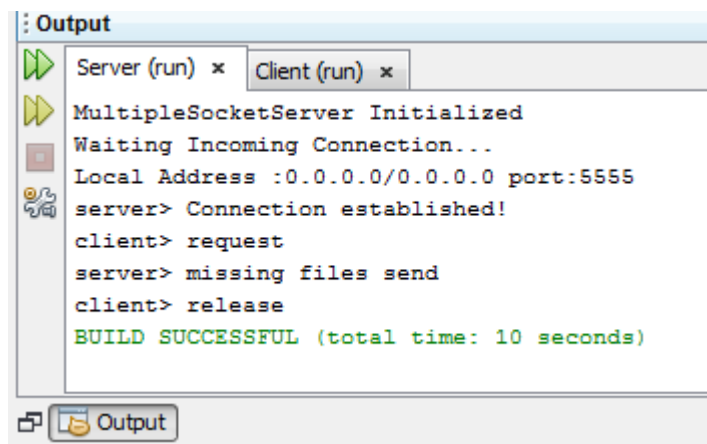
a)Η τεχνολογία που χρησιμοποιήσαμε είναι τα **Socket** για την επικοινωνία των κόμβων.Η τεχνολογία των **Socket** μας φάνηκε πιο εύχρηστη σε σύγκριση με τα **RMI** ως προς την υλοποίηση της άσκησης διότι είναι πιο ευέλικτα και επαρκή. Επιλέξαμε την χρήση sockets για να έχουμε αυστηρότερο έλεγχο σε ό, τι έχει αποσταλεί και μπορούμε να βελτιστοποιήσουμε τα streams, εξουδετερώνοντας ή συμπιέζοντας την ημερομηνία. Είναι πιο εύκολο να εκτελέσουμε το πρόγραμμα από ότι στα RMI που πρέπει να το κάνουμε από το cmd .

b)Ο αλγόριθμος αμοιβαίου αποκλεισμού που χρησιμοποιήσαμε είναι ο **αλγόριθμος κεντρικού συντονιστή**.Βάση αυτόν τον αλγόριθμο μια διεργασία εκλέγεται ως συντονιστής Pc και ελέγχει την πρόσβαση στην ΚΠ.Ο συντονιστής διατηρεί μια ουρά αναβλημένων αιτήσεων.Στο παράδειγμά μας ο client που θέλει πρόσβαση στα αρχεία του server στέλνει μήνυμα request στο server.Ο server λαμβάνει το request εν συνεχεία ο client στέλνει τα αρχεία που λείπουν στον server και ο server τα λαμβάνει.Μετά ο client στέλνει ένα release και ο server το λαμβάνει. Χρησιμοποιήσαμε τον αλγόριθμο αυτό διότι είναι πιο εύχρηστος με την πιο απλή υλοποίηση.

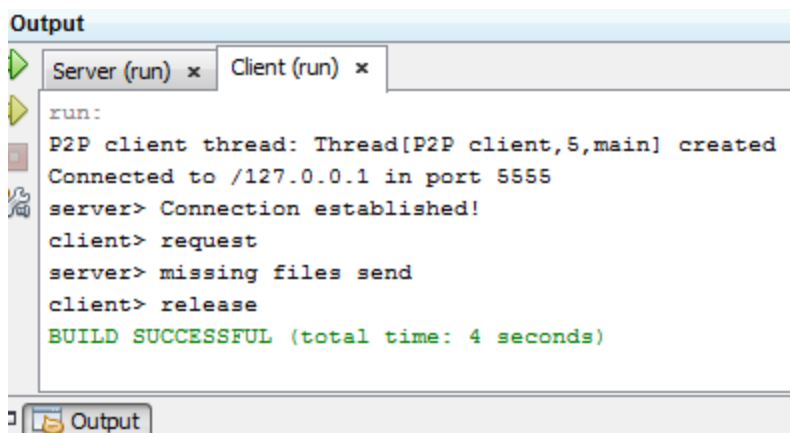
c) Για τον συγχρονισμό έχουμε πρόβλημα στην ανταλλαγή αρχείων μεταξύ server-client αφού θέλουμε να αποτρέψουμε την ταυτόχρονη αποστολή των αρχείων που λείπουν από τον server στον client. Κάνουμε χρήση συγχρονισμού μέσα στο πρόγραμμα και στον server και στον client.

d)Για την υλοποίηση της εφαρμογής φτιάξαμε έναν client και έναν server που επικοινωνούν μέσω socket.Αυτοί ανταλλάσσουν μεταξύ τους αρχεία τα οποία τα έχουμε αποθηκεύσει μέσα σε λίστες.Για την ανταλλαγή αυτή κάναμε χρήση του αλγόριθμου αμοιβαίου αποκλεισμού του **κεντρικού συντονιστή**.Μέσω συγχρονισμού αποτρέψαμε την ταυτόχρονη αποστολή αρχείων που λείπουν.

f) Τρέχω πρώτα τον server και μετά τον client και έχω τα παρακάτω



```
Output
Server (run) x Client (run) x
MultipleSocketServer Initialized
Waiting Incoming Connection...
Local Address :0.0.0.0/0.0.0.0 port:5555
server> Connection established!
client> request
server> missing files send
client> release
BUILD SUCCESSFUL (total time: 10 seconds)
```



```
Output
Server (run) x Client (run) x
run:
P2P client thread: Thread[P2P client,5,main] created
Connected to /127.0.0.1 in port 5555
server> Connection established!
client> request
server> missing files send
client> release
BUILD SUCCESSFUL (total time: 4 seconds)
```

B)Σύγκριση RMI & Socket

Ποια είναι τα πλεονεκτήματα και ποια τα μειονεκτήματα των Socket και του RMI. Εξηγήστε το σκεπτικό με το οποίο επιλέξατε την κάθε τεχνολογία στην υλοποίησή σας.

Απάντηση:

Socket

Πλεονεκτηματα Socket

Τα **sockets** είναι ευέλικτα και επαρκή. Τα αποδοτικά socket βασίζονται στο προγραμματισμό και μπορούν να εφαρμοστούν για γενικές επικοινωνίες. Τα socket προκαλούν χαμηλή κίνηση στο Διαδύκτιο. Σε αντίθεση με τις φόρμες HTML και CGI scripts που δημιουργούν και μεταφέρουν ολοκληρές web σελίδες για κάθε νέα αίτηση, τα Java applets μπορούν να στείλουν μόνο αναγκαίες ενημερωμένες πληροφορίες.

Μειονεκτηματα Socket

Οι περιορισμοί ασφαλείας είναι μερικές φορές καταθλιπτικές επειδή ένα java applet που τρέχει σε ένα πρόγραμμα περιήγησης στο Web είναι μόνο σε θέση να εγκαταστήσει συνδεσεις με την μηχανή από όπου προήλθε και πουθενά αλλού στο Διαδίκτυο. Παρόλα αυτά τα χρήσιμα και βοηθητικά χαρακτηριστικά της java ,τα Socket που βασίζονται στις επικοινωνίες επιτρέπουν μόνο να σταλούν πακέτα ακατέργαστων δεδομένων μεταξύ των εφαρμογών. Τόσο η πλευρά του **client** όσο και η πλευρά του **server** πρέπει να παρέχουν μηχανισμούς για να κάνουν τα δεδομένα χρήσιμα με οποιονδήποτε τρόπο. Δεδομένου ότι οι μορφές δεδομένων και τα πρωτόκολλα παραμένουν ειδικά για εφαρμογές η επαναχρησιμοποίηση των socket που βασίζεται σε εφαρμογές είναι περιορισμένη.

RMI

Η κεντρική ιδέα πίσω από αυτή την τεχνολογία είναι η δυνατότητα να καλέσει τις μεθόδους ενός απομακρυσμένου αντικειμένου, προστατεύοντας τον προγραμματιστή από τον χειροκίνητο χειρισμό των Socket προωθώντας παράλληλα μια καλύτερη αρχιτεκτονική λογισμικού.

Πλεονεκτήματα του RMI

Ένα κύριο πλεονέκτημα είναι η απλότητα και καθαρότητα μιας εφαρμογής, οδηγώντας σε πιο διατηρήσιμες, εύρωστες και ευέλικτες εφαρμογές. Σε σύγκριση με τα Sockets έχουμε λιγότερα καθήκοντα στην εφαρμογή αφού έχουν τη δυνατότητα να μειώσουν σε μεγάλο βαθμό κώδικα και πιο πολύπλοκα συστήματα μπορούν να κατασκευαστούν με σχετική ευκολία. Τα μεγαλύτερα οφέλη δεν περιστρέφονται γύρω από την ευκολία στη χρήση, όμως. Τα RMI μας επιτρέπουν να δημιουργήσουμε ένα κατακευματισμένο σύστημα, ενώ την ίδια στιγμή να αποσυνδέουμε αντικείμενα των client / server.

Ακόμα στα RMI δεν χρειάζεται εγκατάσταση πελάτη, μόνο μια Java με δυνατότητα προγράμματος περιήγησης (ή ένα JRE για εφαρμογές). Όλα τα τμήματα διανέμονται εύκολα, και οι ομάδες ανάπτυξης μπορούν να εργαστούν σε μια "ενότητα" της κατακευματισμένης αρχιτεκτονικής.

Μειονεκτήματα του RMI

Το RMI είναι ελαφρώς λιγότερο αποτελεσματικό από ό, τι τα Sockets λόγω των πρόσθετων που εμπλέκονται στο «στρώμα» και επειδή πρέπει να ασχοληθούν με το μητρώο, προκειμένου να επικοινωνήσουν. Μια άλλη ανησυχία είναι η δημιουργία πολυνηματικών servers με ασφάλεια. Ένα κοινό λάθος είναι να υποθέσουμε ότι το προεπιλεγμένο threading θα επιτρέψει να αγνοηθεί ο κωδικός που εξασφαλίζει στον server μας ένα ασφαλές και ισχυρό νήμα.

Συγκρίνοντας RMI με Sockets

Τα RMI είναι στην πραγματικότητα ένα στρώμα αφαίρεσης χτισμένο πάνω Sockets.

Χρησιμοποιεί Sockets για να κοινοποιούν τα στοιχεία μέσω του δικτύου.

Επιτρέπει τις επίμονες συνδέσεις.

Χρησιμοποιεί Serialization για τη μεταφορά αντικειμένων σε streams.

Επιλέξαμε την χρήση sockets για να έχουμε αυστηρότερο έλεγχο σε ό, τι έχει αποσταλεί και μπορούμε να βελτιστοποιήσουμε τα streams, εξουδετερώνοντας ή συμπιέζοντας την ημερομηνία. Είναι πιο εύκολο να εκτελέσουμε το πρόγραμμα από ότι στα RMI που πρέπει να το κάνουμε από το cmd .