Κατανεμημένα Συστήματα
Επικοινωνία Client/Server
Απομακρυσμένη Κλήση Διαδικασιών

Χάρης Μανιφάβας
Τμήμα Εφ. Πληροφορικής & Πολυμέσων
ΤΕΙ Κρήτης

Υλοποίηση RPC

- Προκειμένου να επιτευχθεί διαφάνεια στην κλήση RPC, εισάγεται η έννοια της ψευδορουτίνας (stub) στην πλευρά τόσο του client όσο και του server
- Η ύπαρξη και λειτουργία του δικτύου κρύβεται από ένα RPC communication package, (γνωστό και ως *RPC Runtime)* το οποίο εκτελείται στους client και server

- Η υλοποίηση λοιπόν του μηχανισμού RPC βασίζεται στα εξής μέρη (components):
  - client
  - client stub
  - RPC Runtime
  - server stub
  - server

Υλοποίηση RPC

- Υλοποίηση ψευδορουτινών
  - Οι διαφορές μεταξύ ψευδορουτινών για διαφορετικές διαδικασίες είναι βασικά στο interface
  - Interface: διαδικασίες που καλούνται από τον πελάτη και υλοποιούνται από τον εξυπηρετητή

- Γλώσσα ορισμού διεπαφών (IDL)
  - Τυποποιημένη περιγραφή παραμέτρων διαδικασιών
  - Συνήθως ανεξάρτητη από γλώσσες προγραμματισμού
  - Αυτόματη δημιουργία κορμών/ψευδορουτινών από την IDL
  - Χρήση βιβλιοθήκης για επικοινωνία και πρόταξη
  - Σύνδεση κορμών με πελάτη και εξυπηρετητή
  - Π.χ. SUN XDR, DCE RPC

Υλοποίηση RPC

- Client: μία διεργασία η οποία εκτελεί την κλήση της απομακρυσμένης διαδικασίας

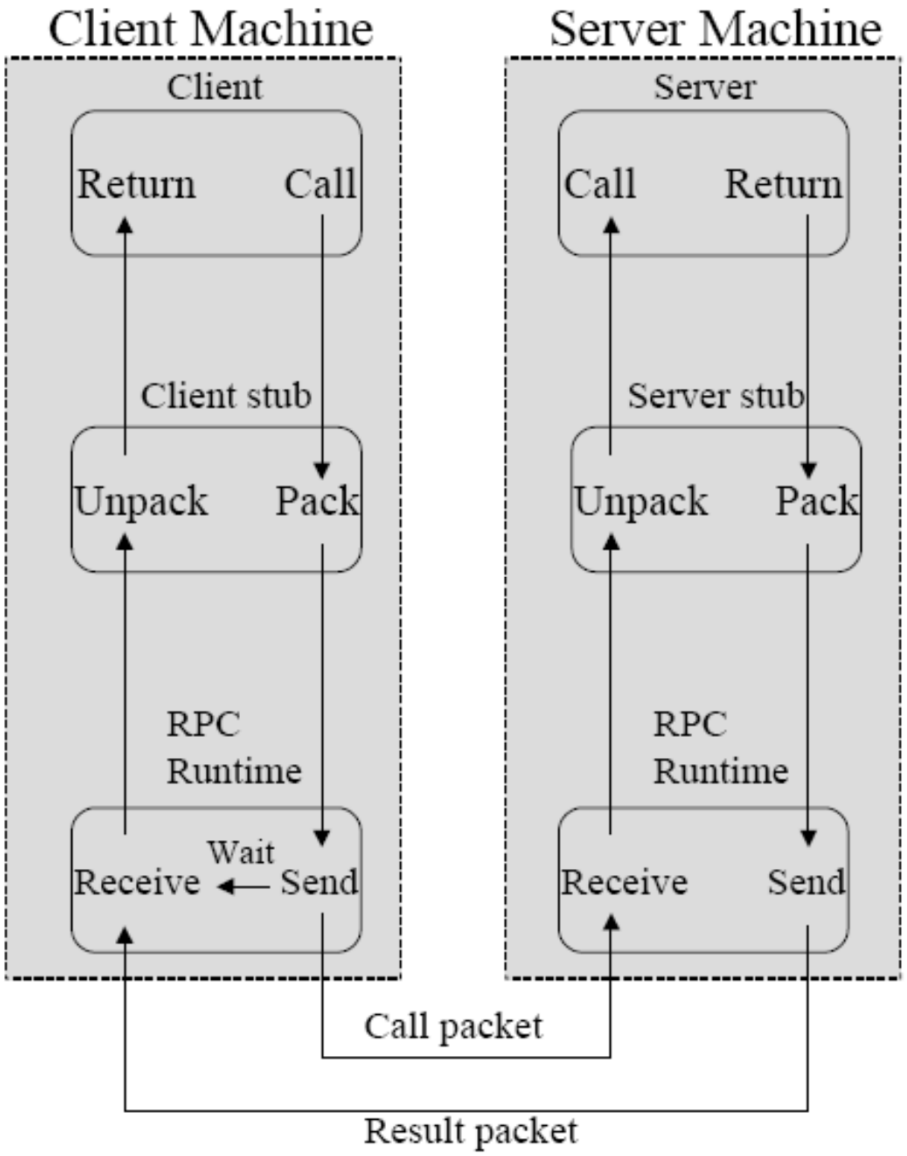
- Client stub: η ψευδορουτίνα που μόλις λάβει την κλήση από τον client πακετάρει την πληροφορία καθορισμού της απομακρυσμένης υπορουτίνας και διευθετεί τα ορίσματα σε ένα μήνυμα
- καλεί το RPC Runtime να στείλει το μήνυμα
- μόλις παραλάβει την απάντηση, την ξεπακετάρει και την παραδίδει στον client

- RPC Runtime: αναλαμβάνει το πρωτόκολλο επικοινωνίας RPC. Είναι υπεύθυνο για retransmissions, acknowledgements, encryption κλπ

Υλοποίηση RPC

- Server stub: η ψευδορουτίνα που μόλις λάβει την κλήση από τον client μέσω του τοπικού RPC Runtime
  - ξεπακετάρει την πληροφορία και εκτελεί την κλήση στην κατάλληλη - τοπική πλέον - διαδικασία στον server
  - μόλις λάβει τα αποτελέσματα της εκτέλεσης από τον server, την πακετάρει (με διευθέτηση δεδομένων) και καλεί το RPC Runtime να στείλει το μήνυμα
- Server: Η διαδικασία η οποία μόλις λάβει την κλήση από το server stub εκτελεί την κατάλληλη διαδικασία και επιστρέφει τα αποτελέσματα

Υλοποίηση RPC

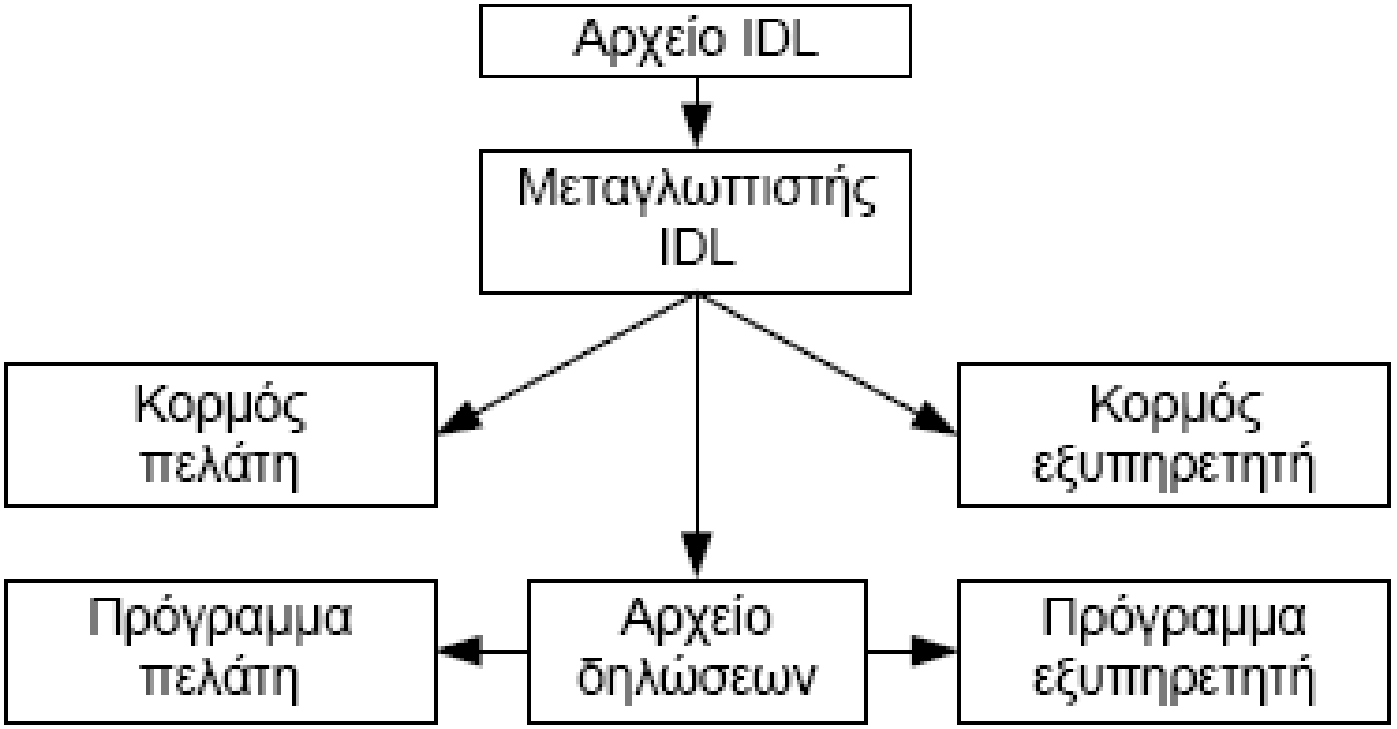


RPC στο Σύστημα DCE

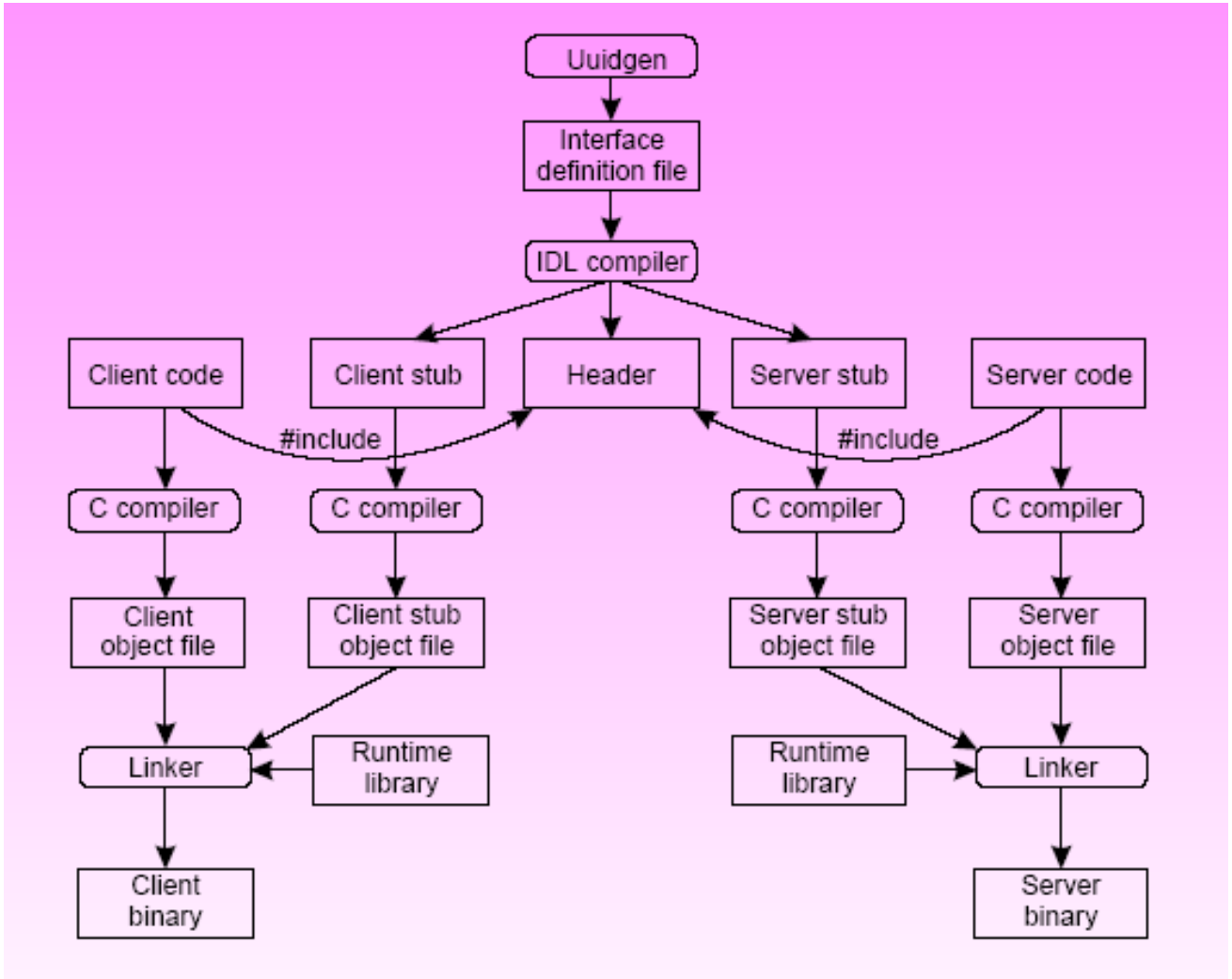
- Κατανεμημένο περιβάλλον επεξεργασίας (DCE)
  - Μοντέλο πελάτη-εξυπηρετητή με ενδιάμεσο λογισμικό
  - Παρέχονται υπηρεσίες DCE και υπηρεσίες χρήστη
  - Όλη η επικοινωνία γίνεται μέσω RPC
  - Αυτόματη μετατροπή τύπων δεδομένων
  - Κατάλληλο για διάφορες γλώσσες προγραμματισμού

- Περιγραφή διεπαφών σε IDL
  - Μοναδικό αναγνωριστικό για τη διεπαφή
  - Περιγραφή παραμέτρων διαδικασιών
  - Περιγραφή σταθερών και τύπων
  - Απεικόνιση στην επιθυμητή γλώσσα προγραμματισμού

RPC στο Σύστημα DCE



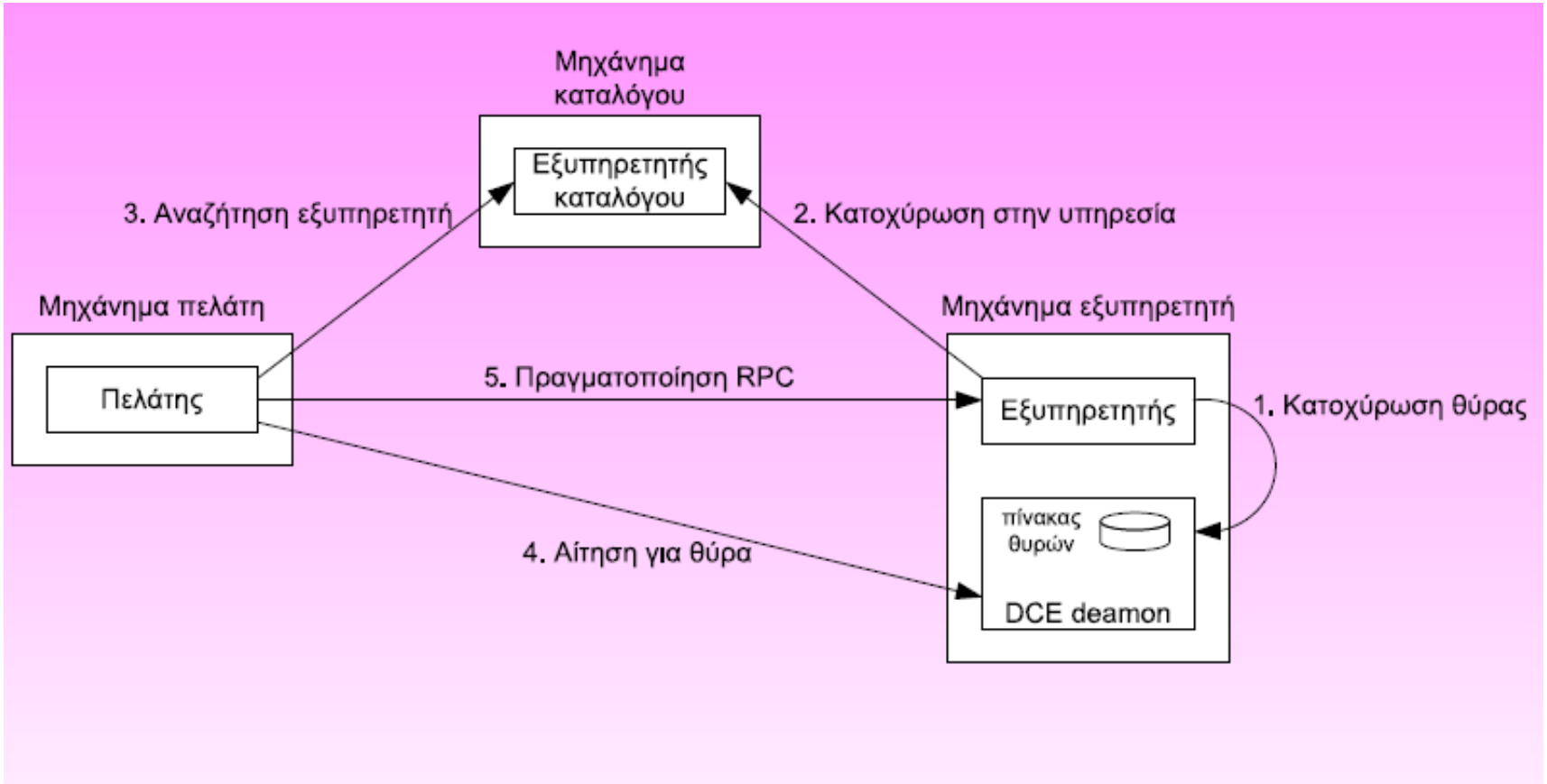
RPC στο Σύστημα DCE



RPC στο Σύστημα DCE

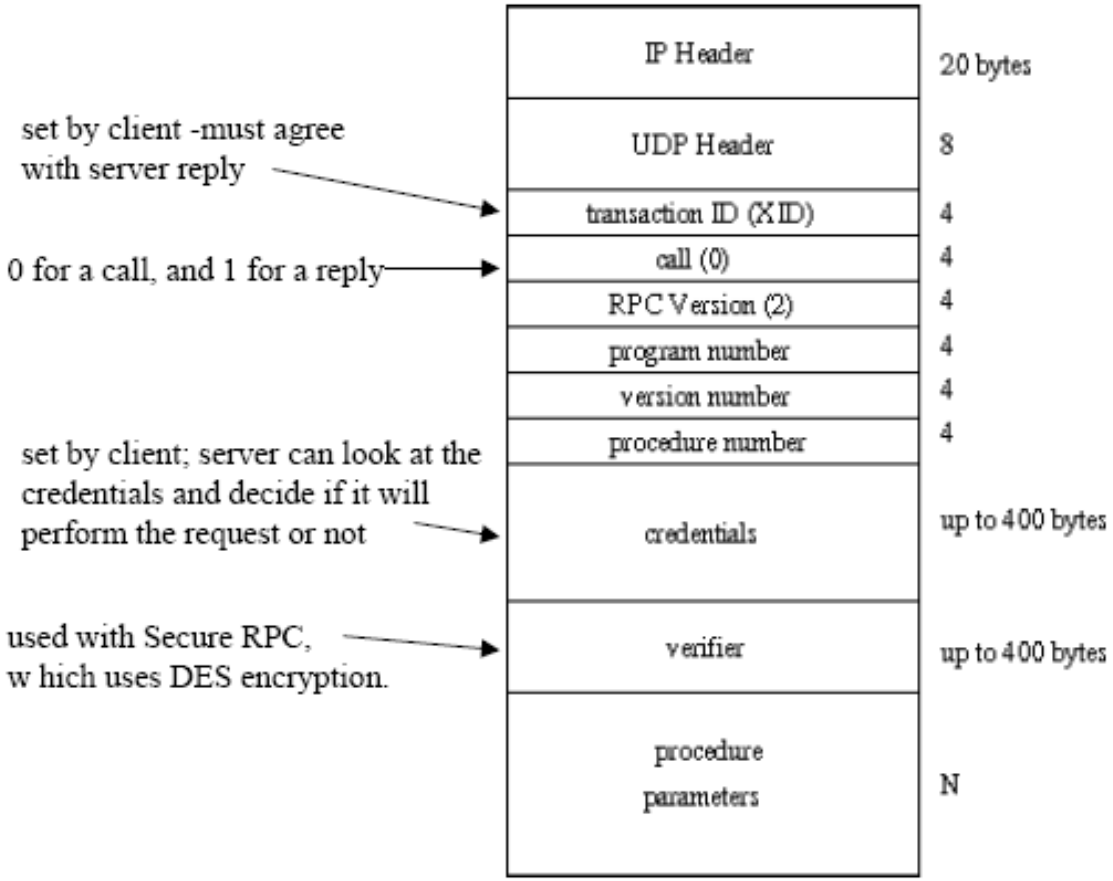
- Επικοινωνία πελάτη και εξυπηρετητή
  - Ο εξυπηρετητής εγγράφεται στην υπηρεσία ευρετηρίου
  - Καταχωρείται το όνομα και η διεύθυνση του εξυπηρετητή
  - Ο εξυπηρετητής εγγράφεται στην τοπική υπηρεσία θυρών
  - Καταχωρείται η θύρα στην οποία αναμένει ο εξυπηρετητής
  - Ο πελάτης εντοπίζει τη διεύθυνση και θύρα του εξυπηρετητή
  - Η διεύθυνση της υπηρεσίας ευρετηρίου πρέπει να είναι γνωστή (πχ fixed ή μέσω αρχικού broadacst)
  - Ο server μπορεί να διαγραφεί από την υπηρεσία ευρετηρίου όταν δεν επιθυμεί να προσφέρει πλέον υπηρεσίες

RPC στο Σύστημα DCE



RPC στο Σύστημα SUN

- Καθορίζει format για μηνύματα, παραμέτρους, αποτελέσματα
- Παράδειγμα μηνύματος RPC με UDP



RPC στο Σύστημα SUN

- Μία συλλογή απομακρυσμένων διαδικασιών αποτελεί ένα *πρόγραμμα*
- Κάθε πρόγραμμα έχει ένα μοναδικό νούμερο (στον server) και έναν αριθμό έκδοσης (version number)
- Κάθε διαδικασία του προγράμματος έχει και αυτή ένα νούμερο (μοναδικό εντός προγράμματος)
- Μία διαδικασία σε ένα server καθορίζεται μοναδικά από: {program-number, version-number, procedure-number}
- Υπάρχει αυτόματη *null procedure* με νούμερο 0, η οποία δεν δέχεται ορίσματα, ούτε επιστρέφει τίποτε, αλλά χρησιμοποιείται ως “ping” για έλεγχο λειτουργίας RPC service στον host
- Τα νούμερα προγραμμάτων είναι:

0x00000000 – 0x1FFFFFFF	Ορίζονται από τη SUN
0x20000000 – 0x3FFFFFFF	Ορίζονται από τον προγραμματιστή/sysadmin
0x40000000 – 0x5FFFFFFF	Μεταβατικά νούμερα, πχ για callback RPC
0x60000000 – 0xFFFFFFFF	Κρατημένα

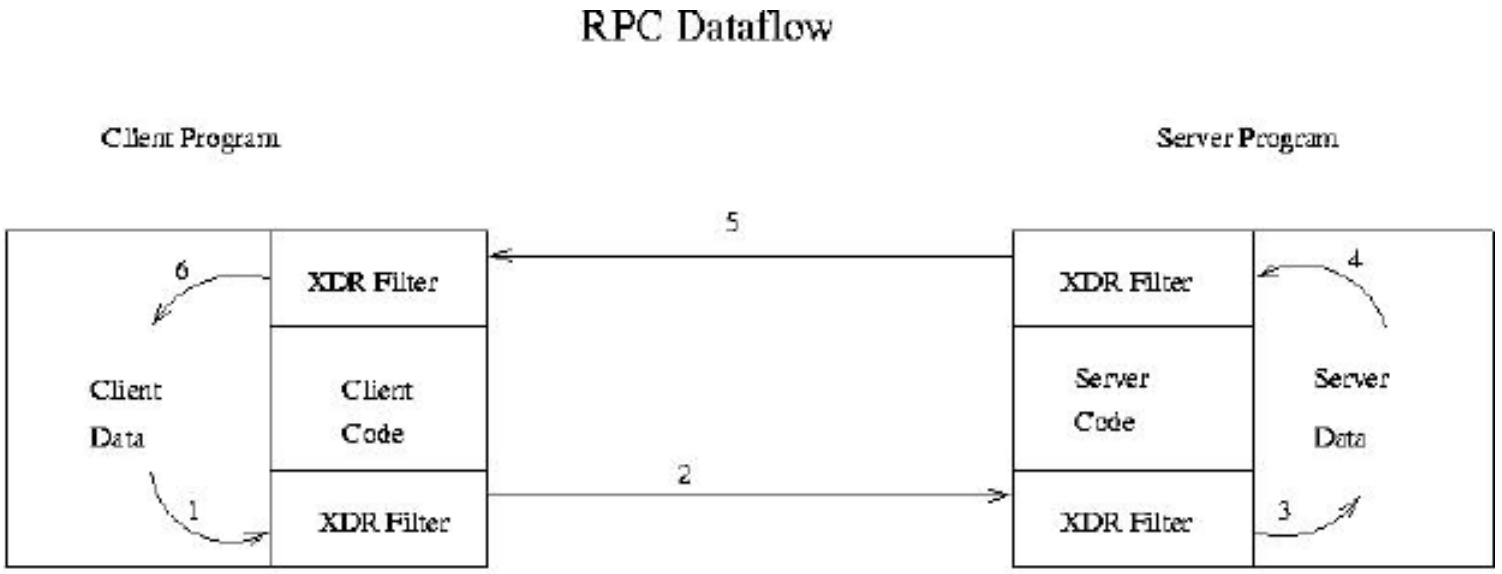
RPC στο Σύστημα SUN

- Ορίζεται ένα timeout, και ένας συνολικός χρόνος αναμονής total_time (default 5 και 25 αντίστοιχα)
  - Μετά την πρώτη εκπομπή, ο client περιμένει απάντηση για χρονικό διάστημα ίσο με timeout
  - Αν δεν λάβει απάντηση, επανεκπέμπει όσες φορές χρειαστεί, έως ότου εκπνεύσει ο total_time
  - Αν εκπνεύσει, τότε το RPC Runtime επιστρέφει timeout error
- Χρησιμοποιεί XDR για ορίσματα, αποτελέσματα και header data

RPC στο Σύστημα SUN

- Χρησιμοποιεί ένα binder (rpcbind deamon) σε κάθε server μηχάνημα (στα well-known ports 111/tcp, 111/udp)
  - Με *rpcinfo* παίρνουμε κατάλογο όλων των registered προγραμμάτων από τον binder
- Ο binder είναι και αυτός ένα RPC πρόγραμμα με νούμερο 100000, version=2
  - Ο binder ξεκινά πριν από όλους τους servers
  - Κάθε RPC server στην εκκίνησή του συνδέεται με ένα socket (ή δημιουργεί νέο) και κάνει registration στον rpcbind (μέσω ενός RPC call) τα
    - program number, version number,
    - και socket (protocol, port number)
  - Ο RPC client στέλνει τα program number, version number, και protocol στον binder και μαθαίνει το port number του server
  - Κατόπιν ο client στέλνει RPC call message στο socket του server, ο οποίος εκτελεί την κατάλληλη διαδικασία

RPC στο Σύστημα SUN



1. Client encodes data through XDR filter.
2. Client passes XDR encoded data accross network to remote host
3. Server decodes data through XDR filter.
4. Server encodes function call result through XDR filter
5. Server pass XDR encoded data accross network back to client
6. Client decodes RPC result through XDR filter and continues processing

Figure 1.

RPC στο Σύστημα SUN

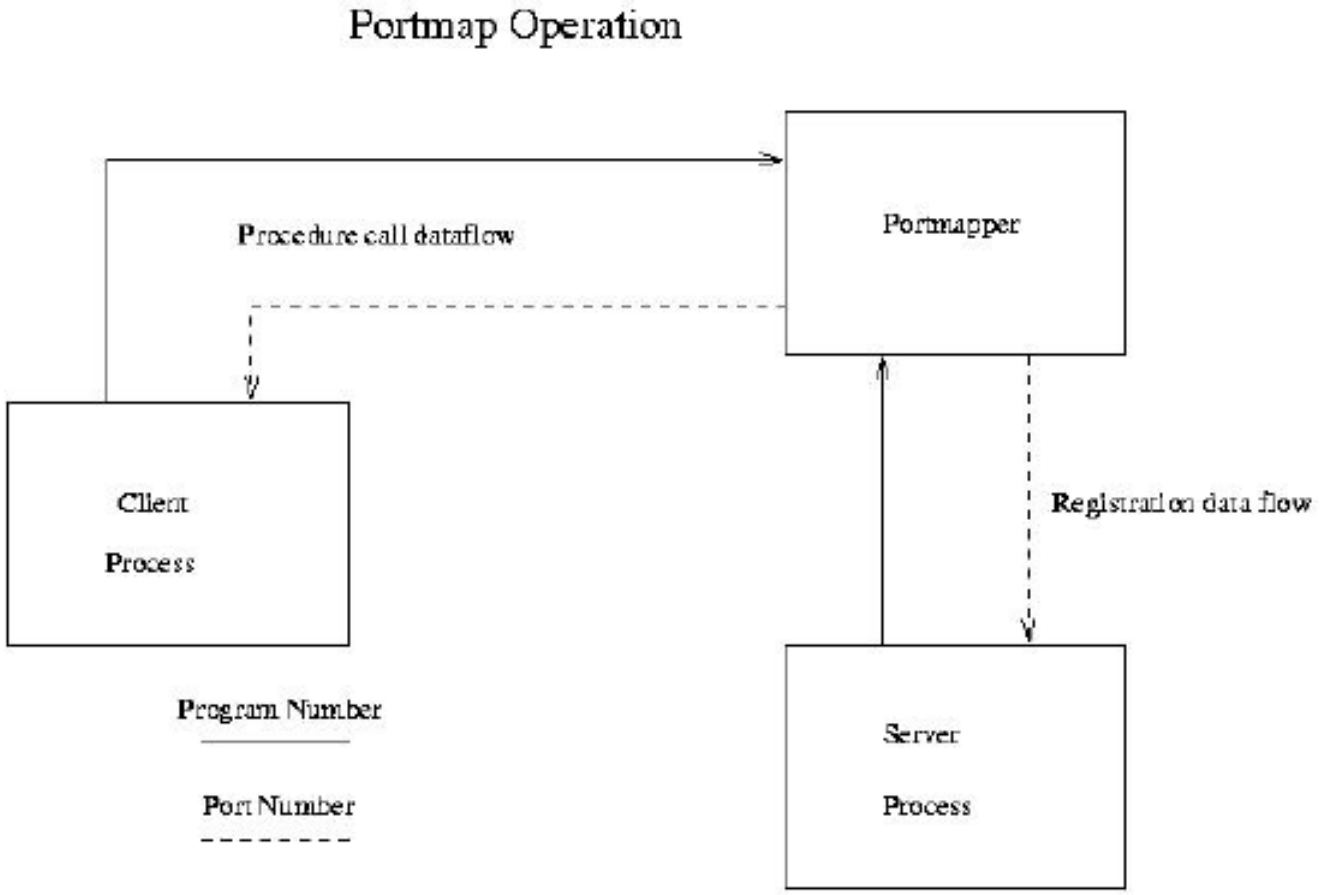
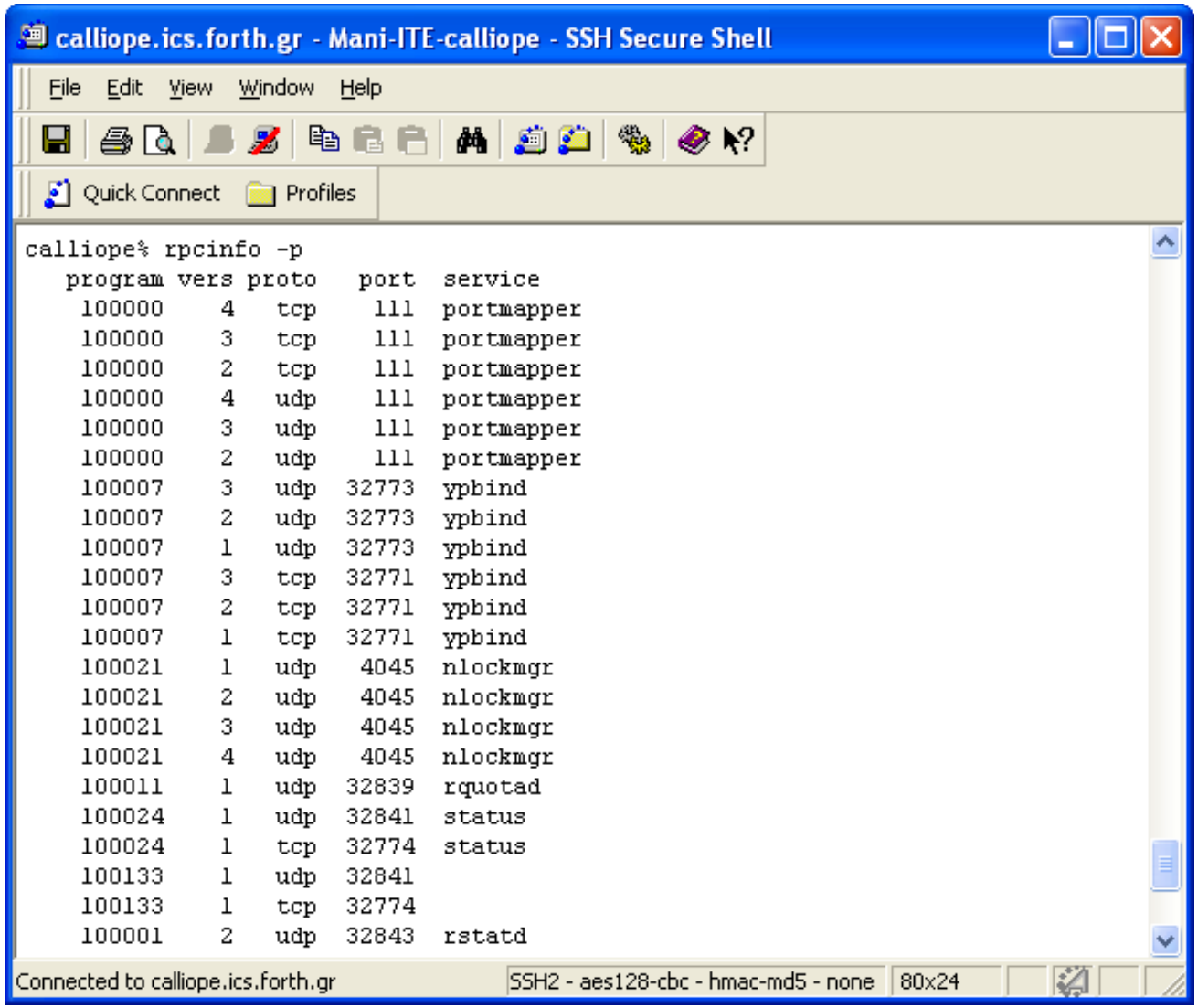


Figure 2

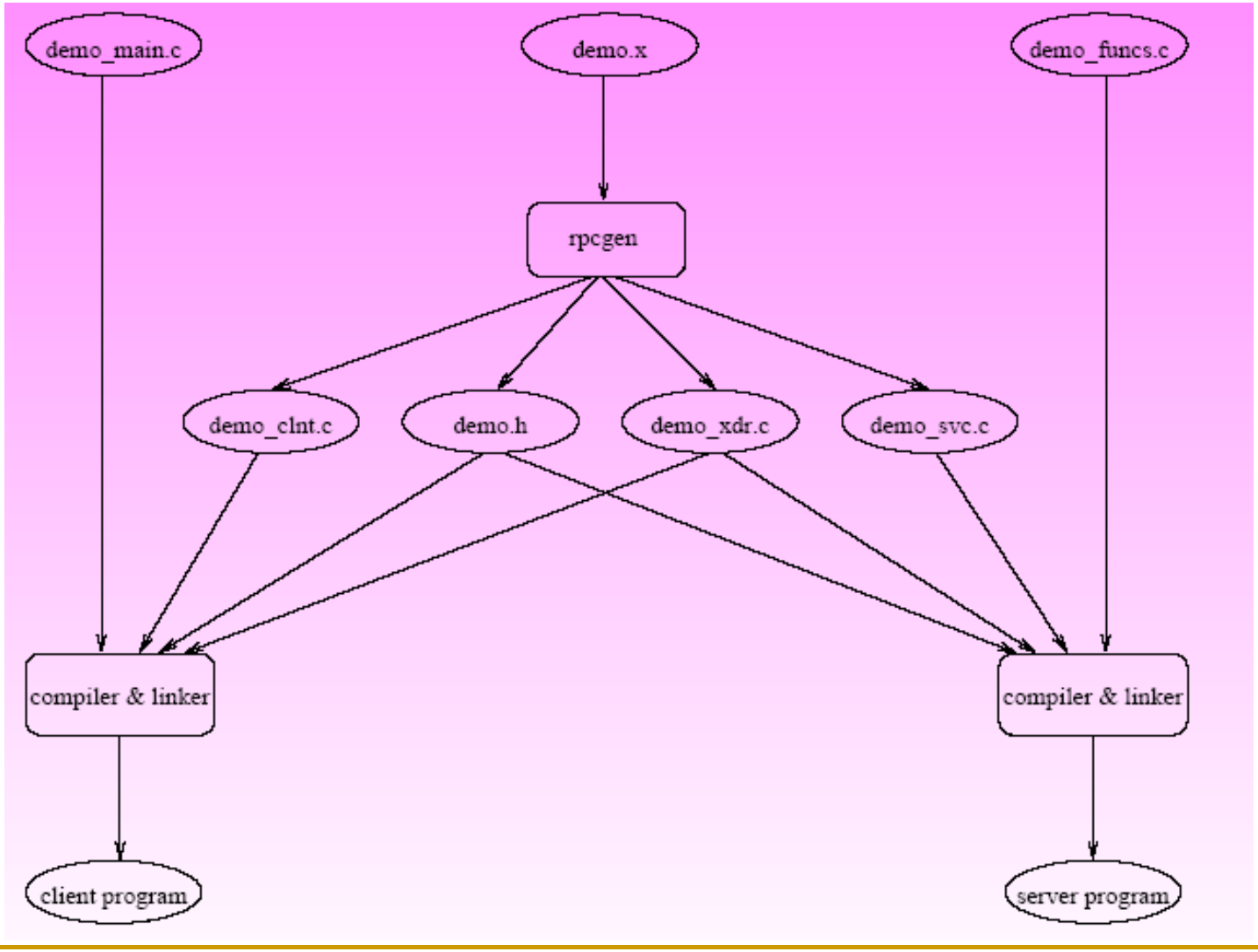
RPC στο Σύστημα SUN



Υλοποίηση SUN RPC

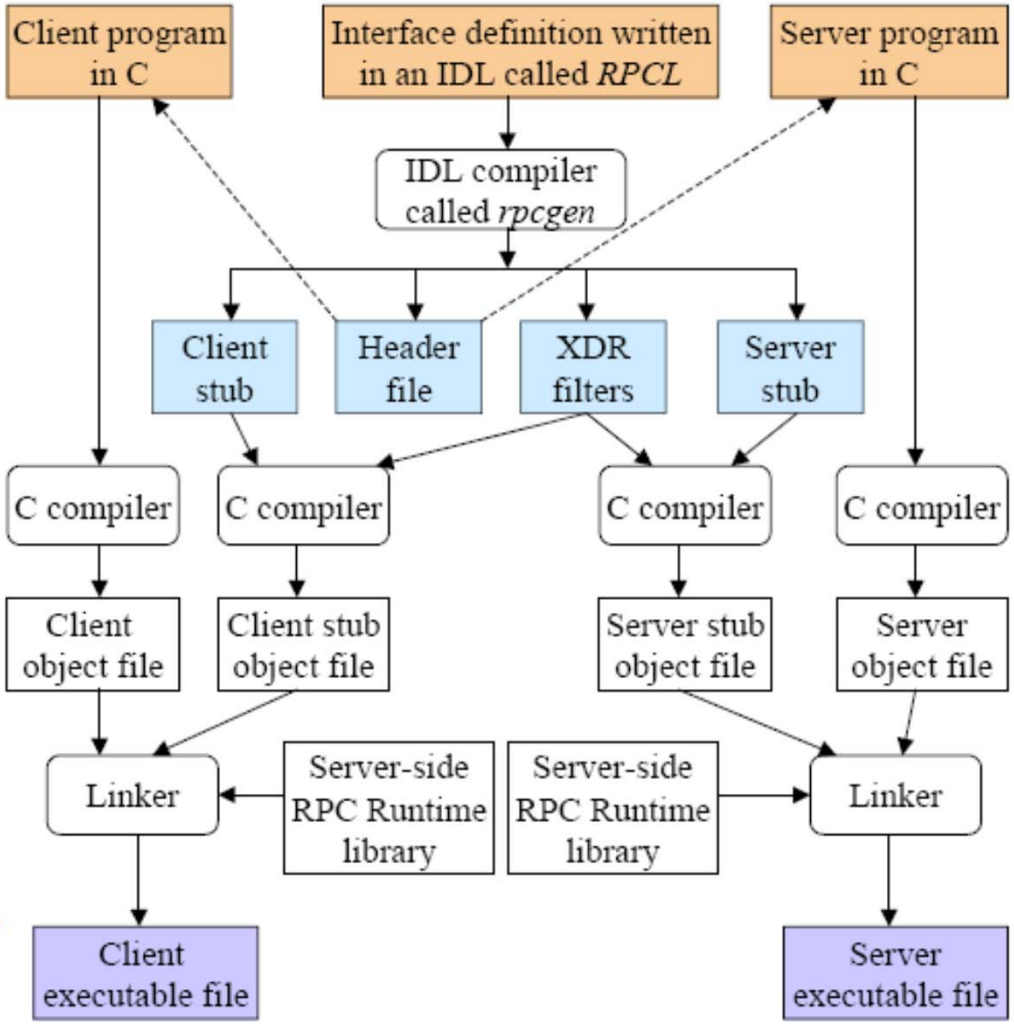
- Απαιτείται η περιγραφή του RPC interface μιας εφαρμογής (ποιες απομακρυσμένες διαδικασίες προσφέρονται και με τι ορίσματα, όνομα προγράμματος και version number κλπ)
- Γράφεται από τον προγραμματιστή σε ένα IDL (interface definition language) που ονομάζεται RPCL και αποτελεί επέκταση του XDR
- Διατίθεται compiler (rpcgen) για αυτόματη δημιουργία stub routines, header file, xdr filter file
- Ο προγραμματιστής γράφει επίσης την απομακρυσμένη διαδικασία, και το πρόγραμμα client

Υλοποίηση SUN RPC



Υλοποίηση SUN RPC

- Αρχεία που γράφει ο προγραμματιστής
- Αρχεία που δημιουργεί αυτόματα ο *rpcgen*



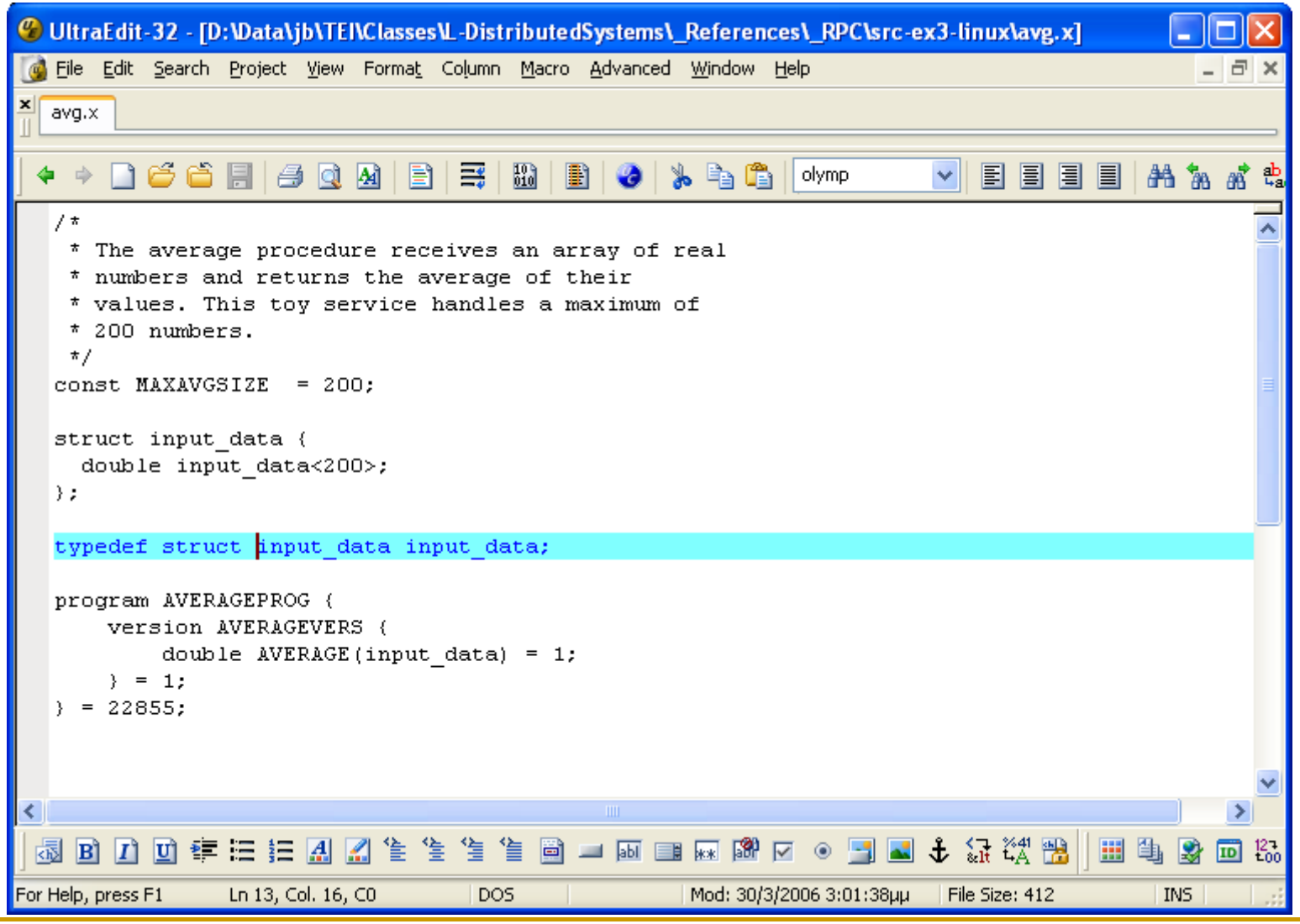
Υλοποίηση SUN RPC

- Μετατροπή Τοπικής Διαδικασίας σε Απομακρυσμένη
  - Έστω το πρόγραμμα (printmsg.c), το οποίο τυπώνει ένα μήνυμα στην κονσόλα Ν φορές, και τρέχει σε έναν Η/Υ
  - Η μετατροπή του σε πρόγραμμα που τυπώνει σε απομακρυσμένους Η/Υ, μέσω RPC απαιτεί την εξής διαδικασία:
    - γράφουμε ένα protocol specification σε γλώσσα RPC (msg.x) το οποίο περιγράφει την απομακρυσμένη έκδοση του `printmessage()'
    - εκτελούμε *rpcgen msg.x* και δημιουργούνται αυτόματα τα αρχεία (αυτόματη δημιουργία ονόματων):
      - msg.h // include file
      - msg_xdr.c //XDR filters
      - msg_clnt.c // client stub
      - msg_svc.c //server stub

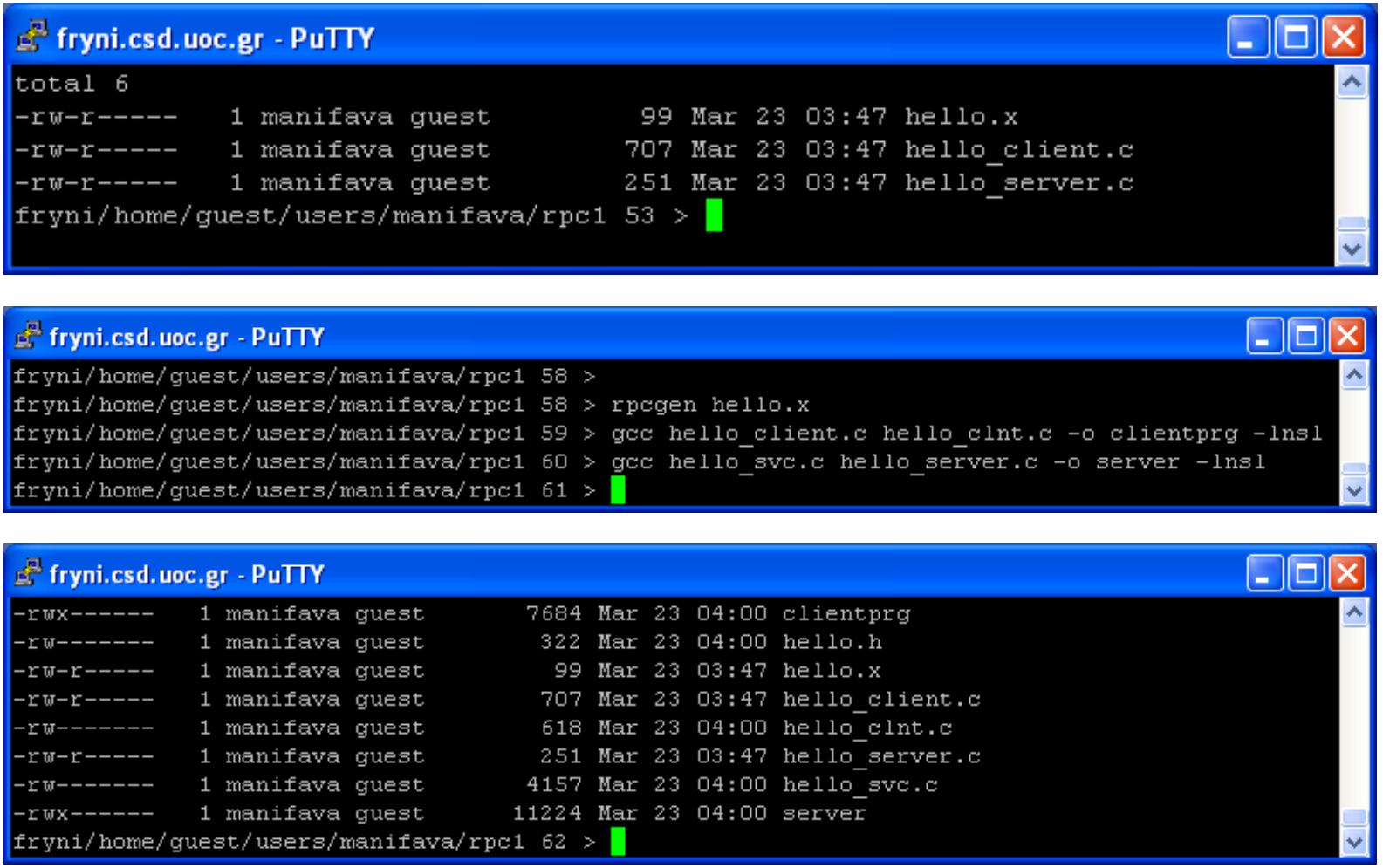
Υλοποίηση SUN RPC

- Μετατροπή Τοπικής Διαδικασίας σε Απομακρυσμένη
  - Στη συνέχεια γράφουμε το client πρόγραμμα rprintmsg.c και το απομακρυσμένο πρόγραμμα msg_proc.c
  - Κατόπιν δημιουργούμε το εκτελέσιμο αρχείο του server:
    - *gcc msg_proc.c msg_svc.c msg_xdr.c –o msg_server*
  - Και το εκτελέσιμο του client:
    - *gcc rprintmsg.c msg_clnt.c msg_xdr.c –o rprintmsg*
  - Αν ο msg_server εκτελείται στο background στη μηχανή hostxxx.epp.teicrete.gr, μπορούμε να τυπώσουμε 10 «Hello» από απομακρυσμένο Η/Υ με την εντολή
    - *rprintmsg* hostxxx.epp.teicrete.gr *“Hello” 10*

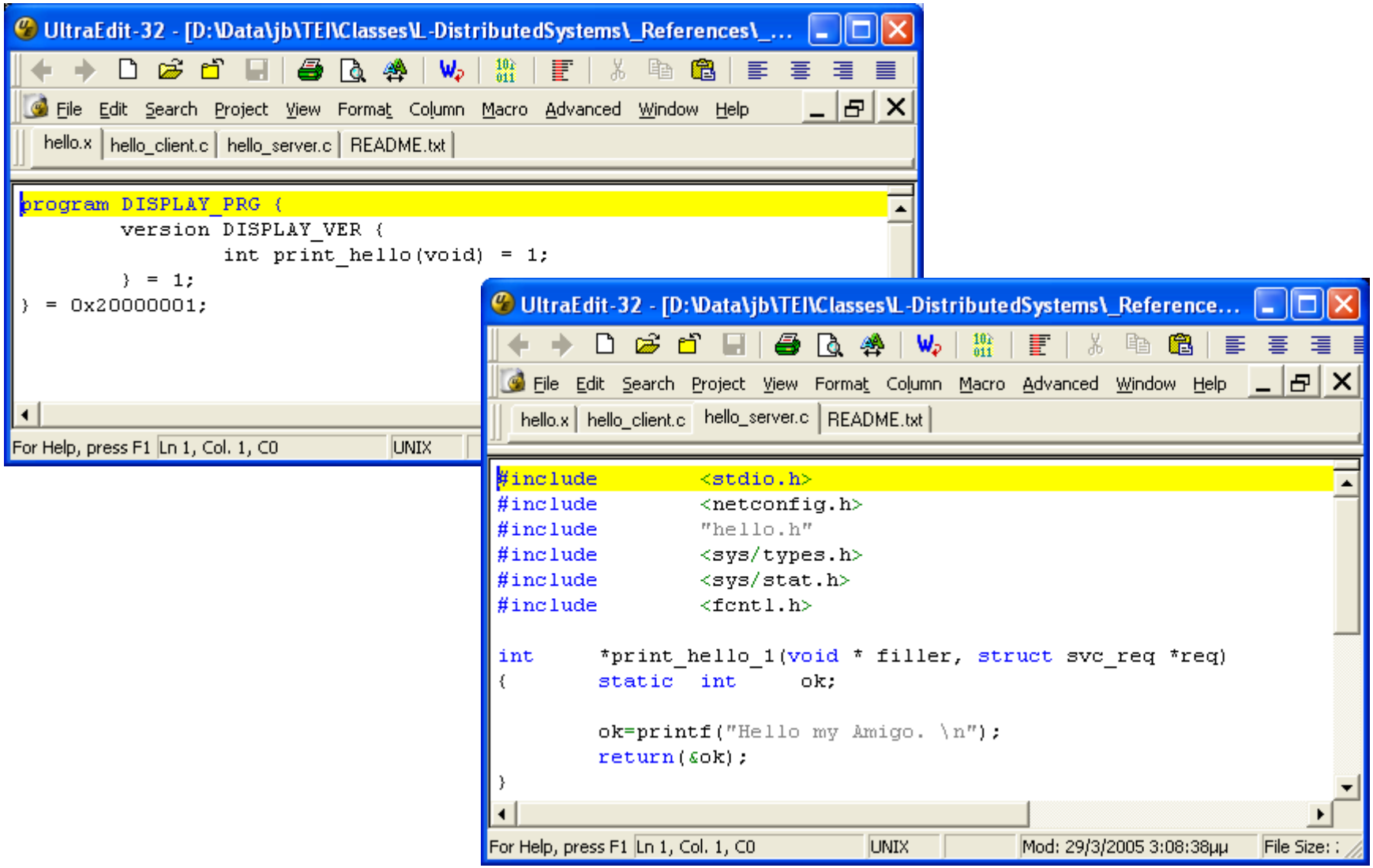
Παράδειγμα RPC - II



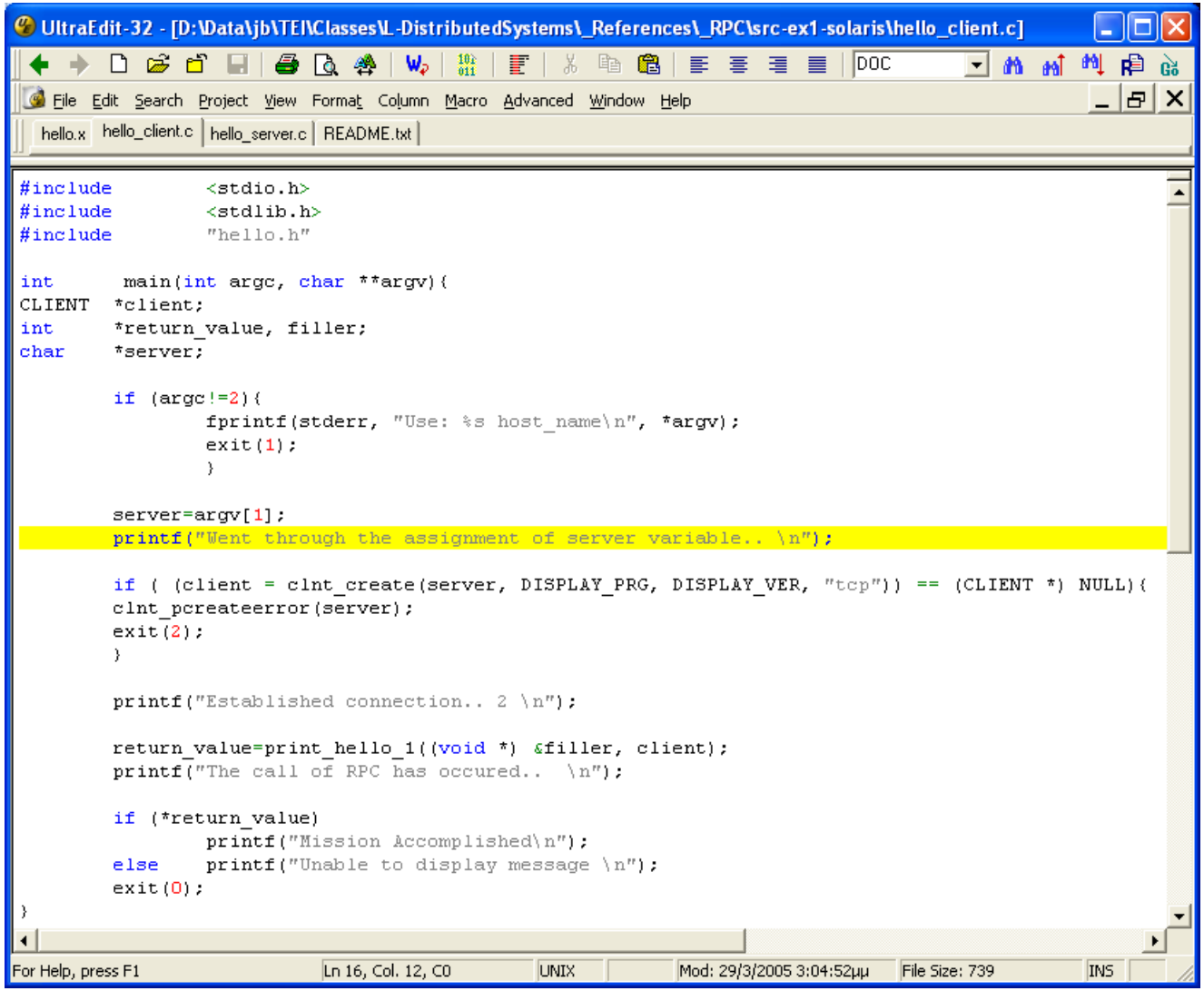
Παράδειγμα RPC



Παράδειγμα RPC



Παράδειγμα RPC



Παράδειγμα RPC