Distributed memory management

Egyed László Attila
Szántó Zoltán
Contents

• Introduction
• Overview
• Components
  – Communication layer
  – Name Server
  – Memory Management Service - MMS
  – Workstation
  – Security Protocols
• Communication Protocols
Introduction

User

I need more !!

Physical memory

1 Mb

User

32 Mb ?!

Memory map

32 Mb

Physical memory

1 Mb

Physical memory

1 Mb

Physical memory

1 Mb

Physical memory

1 Mb
Overview

Memory Management Service 1

Memory Management Service 2

... Memory Management Service n

Work Station

Name Service

Defragmenting Service

Security Service

Message Service with Confirmation
Components – Communication layer

- Request – response
- Use UDP
- Add confirmation
- Add encryption

<table>
<thead>
<tr>
<th>Security Protocols</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message Service with Confirmation</td>
</tr>
<tr>
<td>UDP Message</td>
</tr>
</tbody>
</table>
Components – Communication layer

- Message types:
  - 'No Confirmation'
  - 'Confirmation 2'
  - 'Confirmation 3'

- Security layer
  - Not implemented
Components – Communication layer

- M – sender key
- N – receiver key
- Used with 'Confirmation 3' only

- ID – unique identifier
- TYPE
- DATA
Components – Name Server

- Servers
  - Register
  - Unregister
- Clients
  - Server list
  - Redirect
Components – MMS

- Subscribe to Name Server
- Provide
  - Storage
  - Accessibility:
    - Allocate space or free a memory block
    - Read and Write
- Unsubscribe from Name Server
Components – Workstation

- Used for testing
- Client API
  - Hide internal structure
  - All memory locations seen as one
  - Handle user – system interaction
  - Functions:
    - Allocate or free block
    - Read or Write form a memory block
Components - Security Protocols

- Use with stored data
- Provide data security
- Encryption or ex. access keys
- Left for future development
Communication Protocols

| R/Q | # | TYPE | # | ID | # | .... |

- **MMS**
- **Workstation**
- **Name Server**
Communication Protocols
Name Service – MMS

- Request

<table>
<thead>
<tr>
<th>Q</th>
<th>#</th>
<th>Q_REG_SERVICE</th>
<th>#</th>
<th>ID</th>
<th>#</th>
<th>PORT</th>
<th>#</th>
<th>Service Type</th>
</tr>
</thead>
</table>

- Response

<table>
<thead>
<tr>
<th>R</th>
<th>#</th>
<th>MSG_TYPE</th>
<th>#</th>
<th>ID</th>
</tr>
</thead>
</table>

- Message type:
  - R_REG_SERVICE_OK
  - R_REG_SERVICE_ERR
Communication Protocols
Name Service - Workstation

- Request

| Q | # | Q_GET_SERVICES_LIST | # | ID |

- Response

| R | # | R_SERVICES_LIST | # | ID | # | NR_OF_SERVICES | # | ...... |

- The '....' area

| IP1 | # | PORT1 | # | SERVICE_TYPE1 | # | .. | IPn | # | PORTn | # | SERVICE_TYPEn |
Communication Protocols
MMS – Workstation

• Functions: ALLOC, FREE, WRITE, READ

• Example:
  – Request Q_WRITE_A_BLOCK
  – Response:
    • R_WRITE_BLOCK_OK
    • R_WRITE_BLOCK_ERR

• Extra: Q_GET_NR_OF_FREE_BLOCKS
Thank you!