

Referências Bibliográficas

- [1] GROUP, O. M.. **Common object request broker architecture: Core specification**, 2004. <http://www.omg.org/docs/formal/04-03-01.pdf> (Última Visita em 06/06/2005).
- [2] HENNING, M.; VINOSKI, S.. **Advanced CORBA programming with C++**. Addison-Wesley Longman Publishing Co., Inc., 1999.
- [3] BOLTON, F.; WALSH, E.. **Pure CORBA: A Code-Intensive Premium Reference**. Sams, 2001.
- [4] SOARES, L. F.; LEMOS, G. ; COLCHER, S.. **Redes de Computadores: das LANs, MANs e WANs às Redes ATM**. Campus, 1995.
- [5] GROUP, O. M.. **Corba firewall traversal specification**, 2004. <http://www.omg.org/docs/ptc/04-03-01.pdf> (Última Visita em 06/06/2005).
- [6] RUH, W.; HERRON, T. ; KLINKER, P.. **IIOP complete: understanding CORBA and middleware interoperability**. Addison-Wesley Longman Ltd., 2000.
- [7] BROOKSHIER, D.; GOVONI, D.; KRISHNAN, N. ; SOTO, J. C.. **JXTA: Java P2P Programming**. Sams, 2002.
- [8] GRADECKI, J. D.. **Mastering Jxta: Building Java Peer-to-Peer Applications**. John Wiley & Sons, Inc., 2002.
- [9] HENNING, M.. **A new approach to object-oriented middleware**. IEEE Internet Computing, 8(1):66–75, 2004.
- [10] HENNING, M.; SPRUIELL, M.. **Distributed programming with ice**, 2004. <http://www.zeroc.com/download/Ice/1.4/Ice-1.4.0.pdf> (Última Visita em 06/06/2005).

- [11] TECHNOLOGIES, X. S. I.. **Iiop domain boundary controller - the corba an ejb firewall**, 2004. <http://www.xtradyne.com/documents/whitepapers/Xtradyne-I-DBC-WhitePaper.pdf> (Última Visita em 06/06/2005).
- [12] CERQUEIRA, R.; NOGARA, L. ; MELLO, R.. **Orb o²**, 2004. <http://luaforge.net/projects/o-two/> (Última Visita em 06/06/2005).
- [13] LUA. **The programming language lua**, 2004. <http://www.lua.org> (Última Visita em 06/06/2005).
- [14] IERUSALIMSCHY, R.. **Programming in Lua**. Ingram (US) and Bertram Books (UK), 2003.
- [15] DE MOURA, A. L.; RODRIGUEZ, N. ; IERUSALIMSCHY, R.. **Coroutines in lua**. Journal of Universal Computer Science, 10(7):910–925, July 2004.
- [16] STEVENS, W. R.. **Unix Networking Programming**, volume 1. Prentice-Hall, 2 edição, 1998.
- [17] TOP. **Top - linux man pages**, 2005. <http://www.die.net/doc/linux/man/man1/top.1.html> (Última Visita em 06/06/2005).
- [18] TECHNOLOGIES, I.. **Orbacus**, 2005. <http://www.orbacus.com> (Última Visita em 06/06/2005).
- [19] BROSE, G.. **Jacorb: Implementation and design of a java orb**. Em: PROCS. OF DAIS'97, IFIP WG 6.1 INTERNATIONAL WORKING CONFERENCE ON DISTRIBUTED APPLICATIONS AND INTEROPERABLE SYSTEMS, Cottbus, Germany, October 1997.
- [20] SQUID. **Squid web proxy cache**, 2005. <http://www.squid-cache.org/> (Última Visita em 06/06/2005).

Apêndice A

Arquivo de Configuração

A.1

Document Type Definition – DTD

```
<!ELEMENT firewall-traversal (inbound-path*,outbound-path*,proxy?)>
  <!ATTLIST firewall-traversal choice (omg|proxy) #REQUIRED>
<!ELEMENT inbound-path (element+)>
<!ELEMENT outbound-path (element+)>
<!ELEMENT proxy>
  <!ATTLIST proxy type (tcp|http) #REQUIRED>
  <!ATTLIST proxy address CDATA #REQUIRED>
  <!ATTLIST proxy port CDATA #REQUIRED>
  <!ATTLIST proxy polling-interval CDATA #IMPLIED>
<!ELEMENT element (endpoint+)>
  <!ATTLIST element is-intelligent (true|false) #REQUIRED>
<!ELEMENT endpoint EMPTY>
  <!ATTLIST endpoint service CDATA #REQUIRED>
  <!ATTLIST endpoint address CDATA #REQUIRED>
  <!ATTLIST endpoint port CDATA #REQUIRED>
```

A.2

Exemplo – Abordagem OMG

```
<?xml version='1.0'?>
<!DOCTYPE firewall-traversal PUBLIC "Firewall Traversal
- LAC PUC-Rio" "http://www.lac.inf.puc-rio.br/~theophilo/
firewallTraversal/firewallTraversal.dtd">
<firewall-traversal choice="omg">
  <inbound-path>
    <element is-intelligent="false"> <!-- firewall tcp -->
      <endpoint service="TAG_PASSTHRU_TRANS"
        address="64.82.24.231"
        port="10011"/>
    </element>
    <element is-intelligent="true"> <!-- procurador de aplicacao -->
      <endpoint service="TAG_IIOP_SEC_TRANS"
        address="10.0.0.10"
        port="10011"/>
    </element>
    <element is-intelligent="true"> <!-- ORB Servidor -->
      <endpoint service="TAG_IIOP_SEC_TRANS"
        address="10.0.0.13"
        port="10011"/>
    </element>
  </inbound-path>
  <outbound-path>
    <element is-intelligent="true"> <!-- procurador de aplicacao -->
      <endpoint service="TAG_IIOP_SEC_TRANS"
        address="10.0.0.10"
        port="10011"/>
    </element>
  </outbound-path>
</firewall-traversal>
```

A.3

Exemplo – Abordagem Procurador HTTP

```
<?xml version='1.0'?>
<!DOCTYPE firewall-traversal PUBLIC "Firewall Traversal
- LAC PUC-Rio" "http://www.lac.inf.puc-rio.br/~theophilo/
firewallTraversal/firewallTraversal.dtd">
<firewall-traversal choice="proxy">
  <proxy type="tcp" address="64.82.24.231" port="10012" polling-interval="500" />
</firewall-traversal>
```

Apêndice B

Protocolo da Abordagem Procurador HTTP

Esta seção visa detalhar as mensagens trocadas entre o ORB servidor e o procurador na abordagem Procurador HTTP. A seção B.1 irá apresentar as mensagens enviadas pelo ORB enquanto a seção B.2 irá apresentar as mensagens enviadas pelo procurador. Todas as mensagens apresentadas neste apêndice devem ser antecedidas por seguinte cabeçalho definido em IDL:

```
module HTTPProxy {
    struct HeaderMsg {
        boolean byte_order;
        octet message_type;
        long message_size;
    }
}
```

Este cabeçalho deve ser codificado em CDR e os campos `byte_order`, `message_type` e `message_size` indicam respectivamente a ordenação de bytes utilizadas, o tipo da mensagem e o seu tamanho.

B.1

Mensagens Enviadas pelo ORB Servidor

B.1.1

Register

Requisição de registro de objeto CORBA no procurador. Descrição em IDL:

```
module HTTPProxy {
    const long REGISTER_MSG_ID = 0
    struct Register {
        IOP.IOR ior;
        long orb_id;
    }
}
```

O campo `ior` contém o IOR gerado pelo ORB que será atualizado pelo procurador para conter um par endereço/porta seu. O campo `orb_id` contém o identificador do ORB a fim de que requisições futuras a este objeto possam ser encaminhadas corretamente.

B.1.2

Remove

Requisição de remoção de registro de objeto CORBA no procurador.
Descrição em IDL:

```
module HTTPProxy {
    const long REMOVE_MSG_ID = 2
    struct Remove {
        IOP.IOR ior_exported;
        long orb_id;
    }
}
```

O campo `ior_exported` contém o IOR gerado pelo procurador enquanto que o campo `orb_id` contém o identificador do ORB.

B.1.3

Reply

Mensagem utilizada para enviar ao procurador uma mensagem GIOP Reply.

```
module HTTPProxy {
    const long REPLY_MSG_ID = 4
    struct Reply {
        sequence<octet> reply_message;
        long orb_id;
    }
}
```

O campo `reply_message` contém a mensagem GIOP Reply codificada em CDR enquanto que o campo `orb_id` identifica o ORB.

B.1.4

Polling

Mensagem de consulta por novas requisições armazenadas no procurador.

```
module HTTPProxy {
    const long POLLING_MSG_ID = 5
    struct Polling {
        long orb_id;
    }
}
```

Esta mensagem contém apenas o campo `orb_id` identificando o ORB em questão.

B.2

Mensagens Enviadas pelo Procurador

B.2.1

RegisterReply

Mensagem de resposta a uma requisição de registro de objeto.

```
module HTTPProxy {
    const long REGISTER_REPLY_MSG_ID = 1
    struct RegisterReply {
        IOP.IOR ior_exported;
        long orb_id;
    }
}
```

O campo `ior_exported` contém o IOR atualizado pelo procurador enquanto que o campo `orb_id` contém o identificador do ORB perante o procurador.

B.2.2

Request

Mensagem utilizada para enviar uma mensagem GIOP Request.

```
module HTTPProxy {
    const long REQUEST_MSG_ID = 3
    struct Request {
        sequence<octet> request_message;
    }
}
```

O único campo desta mensagem contém a mensagem GIOP Request codificada em CDR.