

# Segundo trabalho laboratorial de Redes de Computadores (MIEIC)

## 1. Aplicação FTP

Requisitos mínimos:

- Desenvolvimento de uma aplicação baseada num cliente FTP que permita controlar a transferência de ficheiros entre dois servidores FTP
  - Navegação nos servidores remotos (ls, cd, mkdir)
  - Validação da existência do ficheiro (nos servidores de origem e destino)
- Embora descrita no RFC 959, a transferência directa de ficheiros entre dois servidores controlada por um cliente não é actualmente possível na maior parte dos casos, uma vez que os servidores, em geral, não permitem que o endereço enviado no comando PORT (que devia ser o do outro servidor) seja diferente do da máquina que envia o comando (cliente).
- Por esta razão a transferência de um ficheiro entre os servidores deve ser realizada através do cliente, sem ser necessário receber a totalidade do ficheiro do servidor de origem antes do seu envio para o servidor de destino (pode, no entanto, ser feita durante a transferência uma cópia local do ficheiro na máquina onde corre a aplicação cliente) – a ligação com o servidor de origem deve ser estabelecida pelo cliente (modo passivo / comando PASV) e a ligação com o servidor de destino deve ser estabelecida pelo servidor (modo activo / comando PORT).

Valorização:

- Interface gráfica
- Processamento de *wildcards* (\*, ?) para transferência de múltiplos ficheiros

## 2. Cliente POP3 + SMTP

Requisitos mínimos:

- Recepção e envio de mensagens com *attachment(s)* (MIME) e identificação de *body parts*

Valorização:

- Interface gráfica
- Abertura de mensagens recebidas
- Selecção de mensagens com base no cabeçalho, sem necessidade de transferência da mensagem completa (funcionalidade idêntica a IMAP)
- Organização em *folders* (por assunto)

## 3. Cliente IRC

Requisitos mínimos:

- Funcionalidade básica – comandos: HELP, LIST, JOIN, PART, QUIT, NICK, AWAY, WHOIS, INVITE, KICK, TOPIC, ME

Valorização:

- Interface gráfica
- Comandos adicionais: MSG, QUERY, NOTICE, NOTIFY, IGNORE
- Escape para a *shell*

## 4. Agenda electrónica

Requisitos mínimos:

- Agenda individual
  - Consulta
  - Anotações – acrescentar e remover eventos
  - Alarmes associados a eventos – envio de mensagens com antecedência e periodicidade programadas

Valorização:

- Acesso multi-utilizador, com *username / password* e níveis de prioridade para consulta e modificação

## 5. Proxy http

Requisitos mínimos:

- Envio (*relay*) de pedidos para o servidor
- Transferência das páginas solicitadas
- *Cache* de páginas estáticas

Valorização:

- *Logging* das ligações (no ficheiro de *log*, uma linha por página transferida, em formato Apache)
- Ligações seguras https
- Processamento simultâneo de pedidos

## 6. Cliente e servidor HTTP

Requisitos mínimos:

- Implementação de um cliente e um servidor HTTP muito simples
  - O cliente deverá ser capaz de enviar comandos simples HTTP, nomeadamente GET e GET condicional
  - O cliente deverá ser capaz de comunicar directamente com qualquer servidor HTTP
  - O servidor deverá responder a comandos GET e GET condicional
  - O servidor deverá ser capaz de comunicar directamente com qualquer cliente HTTP
  - O servidor deverá ser capaz de comunicar com mais do que um cliente simultaneamente

Valorização:

- Capacidade do cliente comunicar com *proxies* HTTP
- Capacidade do cliente descarregar uma página *web* completa
- Utilização de ligações persistentes

O cliente deverá suportar a seguinte linha de comandos:

```
>./clienteHTTP [-Proxy <pname> -ProxyPort <pport>]
```

em que `-Proxy` é o argumento que sugere a utilização de um *proxy* em vez de comunicar directamente com o servidor, `<pname>` é o *hostname* do *proxy* e `<pport>` é a porta TCP do *proxy*.

Depois de executado, o cliente deverá estar pronto a aceitar comandos **sucessivos** como no exemplo seguinte:

```
GET www.fe.up.pt/~abc/RCOM/index.html {-Port <port>}{-Cond}{-Full}
```

- <port> é a porta do servidor (por omissão deverá ser 80)
- O objecto descarregado deverá ser guardado localmente. Por exemplo para `www.fe.up.pt/~abc/RCOM/index.html` deverá ser criada uma pasta `www.fe.up.pt`, dentro da qual deverá ser criada a pasta `~abc`, dentro da qual deverá ser criada uma pasta `RCOM`, onde finalmente ficará guardado o ficheiro `index.html`.
- Se o parâmetro `-Cond` for usado o cliente deverá enviar um GET condicional com a opção **if-modified-since**. O *timestamp* para o GET condicional será sempre o tempo especificado na opção **Last-Modified** do último acesso ao ficheiro (o cliente tem que guardar o atributo **Last-Modified**).
- Se o parâmetro `-Full` for especificado, o cliente deverá descarregar o documento completo (suportando redireccionamentos, *frames* e imagens).

O servidor deverá suportar a seguinte linha de comandos:

```
>./servidorHTTP -Port <port> -root <folder>
```

em que `-Port` indica a porta em que escuta pedidos e `-root` a pasta raiz em relação à qual os objectos são referenciados. Todos os pedidos devem ser registados (*logfile*).

## Notas gerais

- O núcleo da aplicação deve ser desenvolvido em C; no desenvolvimento de interfaces gráficas ou outros melhoramentos podem ser usadas bibliotecas e linguagens à escolha.
- Para além das aplicações sugeridas, são admitidas outras, desde que seja utilizado um protocolo normalizado e seja possível fazer o teste com um cliente ou um servidor genérico.
- Relativamente aos elementos de valorização sugeridos em cada trabalho, a qualidade é mais importante do que a quantidade.

## **Avaliação**

- Contínua
- Apresentação e demonstração do trabalho
- Relatório

## **Demonstração**

- Nas aulas práticas de 31 de Maio a 4 de Junho de 2010

## **Entrega do Relatório**

- Por *email* para o docente da aula teórico-prática respectiva
- Material a entregar:
  - Relatório com código fonte em anexo
- Data limite: 9 de Junho de 2010