Universidade de Aveiro

Departamento de Eletrónica, Telecomunicações e Informática

Sistemas Operativos

Multi-Threaded Web Server

Manual de Utilizador

Autores:
Diogo Ruvio (NMec: 126498)
David Cálix (NMec: 125043)

Turma/Grupo: P2 / G7

Dezembro 2025

Conteúdo

1 Introdução

O **Multi-Threaded Web Server** é um servidor web de alto desempenho desenvolvido em C, capaz de processar múltiplos pedidos simultaneamente utilizando uma arquitetura *Master-Worker* e *thread pools*.

O sistema implementa funcionalidades avançadas como cache LRU em memória, suporte a CGI, estatísticas em tempo real e *Virtual Hosts*. Este manual descreve os passos necessários para compilar, configurar e operar o servidor em ambiente Linux.

2 Pré-requisitos

Para garantir o funcionamento correto do servidor e da suite de testes, certifique-se que o sistema possui o software listado abaixo.

2.1 Essenciais (Para Executar)

Software necessário para compilar e iniciar o servidor:

- **Sistema Operativo:** Linux (Ubuntu 20.04+ ou similar).
- **Compilador:** GCC (com suporte a biblioteca `pthread`).
- **Build System:** Make.

2.2 Ferramentas de Teste (Recomendado)

Para executar os scripts de teste automatizados (`tests/`), é necessário instalar:

- **curl**: Para realizar pedidos HTTP de teste.
- **apache2-utils**: Contém a ferramenta `ab` para testes de carga.
- **valgrind**: Para deteção de fugas de memória e *race conditions*.

2.2.1 Comando de Instalação Rápida

Pode instalar todas as dependências necessárias no Ubuntu/Debian com o seguinte comando:

```
$ sudo apt-get update
$ sudo apt-get install build-essential curl apache2-utils valgrind
```

3 Compilação

O projeto inclui um `Makefile` para automatizar o processo de compilação.

3.1 Compilar o Projeto

Abra o terminal na raiz do projeto e execute:

```
$ make
```

Isto irá gerar o executável `server` na raiz do projeto.

3.2 Limpar Ficheiros Temporários

Para remover os executáveis, ficheiros objeto (.o) e limpar recursos IPC antigos:

```
$ make clean
```

4 Configuração (server.conf)

O comportamento do servidor é controlado pelo ficheiro `server.conf`. Certifique-se que este ficheiro existe na mesma diretoria do executável.

Parâmetro	Padrão	Descrição
`PORT`	8080	Porta TCP onde o servidor escuta conexões.
`NUM_WORKERS`	4	Número de processos *Worker* a iniciar.
`THREADS_PER_WORKER`	10	Dimensão da *Thread Pool* por Worker.
`DOCUMENT_ROOT`	./www	Diretoria raiz para ficheiros estáticos.
`MAX_QUEUE_SIZE`	100	Capacidade da fila de conexões pendentes.
`CACHE_SIZE_MB`	10	Tamanho máximo da Cache LRU (MB).
`TIMEOUT_SECONDS`	30	Intervalo de atualização de stats e *keep-alive*.
`VHOST_<host>`	N/A	Define a raiz para um *Virtual Host*.
`LOG_FILE`	access.log	Caminho do ficheiro de registo de acessos.

Tabela 1: Parâmetros de Configuração suportados

5 Execução

Para iniciar o servidor com a configuração padrão:

```
$ ./server
```

O servidor iniciará, limpará preventivamente recursos de memória partilhada antigos e apresentará o PID do processo *Master*. Para encerrar o servidor de forma graciosa (*graceful shutdown*), pressione **Ctrl+C**.

6 Como Testar e Usar

6.1 Testes Básicos

Uma vez o servidor em execução, pode testar o funcionamento básico via browser ou terminal:

```
# Pedido simples
$ curl -v http://localhost:8080/index.html
```

6.2 Funcionalidades Avançadas

O servidor suporta várias funcionalidades extra que podem ser testadas da seguinte forma:

Dashboard de Estatísticas:
Aceda a http://localhost:8080/stats para visualizar métricas em tempo real (Uptime, Conexões Ativas, Cache Hits, etc.).

Execução de CGI (Python):
Coloque ficheiros Python (`.py`) na pasta `www`. O servidor executará o script e devolverá o output.

```
$ curl http://localhost:8080/test.py
```

Virtual Hosts:
Para testar sites configurados como *VHosts* (ex: `site1.local` configurado no `server.conf`):

```
$ curl -H "Host: site1.local" http://localhost:8080/
```

7 Resolução de Problemas

- **Erro "Address already in use":** O servidor anterior pode não ter fechado corretamente o socket. Aguarde alguns segundos ou altere a `PORT` no ficheiro de configuração.
- **Erro 403 Forbidden:** Verifique se o ficheiro solicitado tem permissões de leitura no sistema operativo (`chmod +r`).
- **Semáforos Bloqueados:** Se o servidor crashar, recursos IPC podem ficar presos. Execute `make clean` ou `make run` para forçar a limpeza de `/dev/shm`.