User Manual

Concurrent HTTP Web Server

[Student Name 1] (ID: [XXXXX]) Student Name 2>Student Name 2 (ID: [YYYYY])

Operating Systems - University of Aveiro

December 5, 2025

Contents

1 Introduction

This manual provides instructions for compiling, configuring, and operating the concurrent web server developed for the Operating Systems course. The server supports HTTP/1.1 requests, multi-process and multi-thread architecture, and shared memory statistics.

2 System Requirements

To run this server, the environment must meet the following requirements:

- **Operating System:** Linux/Unix (e.g., Ubuntu 20.04 or later).
- **Compiler:** GCC with C99 support.
- **Libraries:** `pthread`, `rt` (Real-time extensions).
- **Tools:** `make`, `curl`, `ab` (Apache Bench for testing).

3 Compilation

The project uses a `Makefile` for build automation. Run the following commands from the project root:

3.1 Standard Build

To build the production executable `bin/webserver`:

```
make all
```

3.2 Cleaning

To remove object files and old binaries:

```
make clean
```

3.3 Debug/Test Modes

- **Helgrind (Race Detection):** `make helgrind`
- **ThreadSanitizer:** `make tsan`

4 Configuration

The server behavior is controlled by the `server.conf` file located in the root directory.

Parameter	Example	Description
PORT	8080	TCP port to listen on.
DOCUMENT_ROOT	www	Directory containing static files.
NUM_WORKERS	4	Number of worker processes to fork.
THREADS_PER_WORKER	10	Number of threads per worker.
MAX_QUEUE_SIZE	100	Size of the shared request buffer.
LOG_FILE	server.log	Path to the access log file.
CACHE_SIZE_MB	10	LRU Cache size per worker (in MB).

Table 1: Configuration Parameters

5 Execution

5.1 Starting the Server

After compilation, start the server with:

```
./bin/webserver
# Or via make:
make run
```

5.2 Graceful Shutdown

To stop the server correctly (releasing shared memory and closing sockets):

1. Press **Ctrl+C** (sends `SIGINT`) in the terminal running the server.
2. The Master process will catch the signal, notify workers, and clean up resources.
3. Wait for the "Server shutdown complete" message.

6 Usage Examples

6.1 Basic HTTP Request

Using `curl` to fetch the index page:

```
curl -v http://127.0.0.1:8080/index.html
```

6.2 Checking Statistics

The server prints statistics to the standard output every 30 seconds. You can also view the logs:

```
tail -f server.log
```

7 Troubleshooting

- **Error: Address already in use**
  Solution: The configured PORT is occupied. Change it in `server.conf` or kill the process using it.

- **Error: Shared Memory / Semaphores failed**
  Solution: If the server crashed previously, old shared memory objects might persist. Delete them from `/dev/shm/` or restart the PC.