

# ConcurrentHTTP Server - User Manual

Sistemas Operativos 2025/2026 - Trabalho Prático 2

Autores: Martim Dias 125925, Nuno Costa 125120

---

## 1. System Requirements

### Required Software:

- **Operating System:** Linux (Ubuntu 20.04 or later or other distributions)
- **Compiler:** GCC 9.0+ with C11 support
- **Libraries:** pthread, rt (POSIX real-time)

### Installation:

```
# Ubuntu/Debian
sudo apt-get update
sudo apt-get install -y build-essential gcc make

# Verify installation
gcc --version
make --version
```

---

## 2. Compilation

### Using Makefile:

```
# Build the server
make

# Clean build artifacts
make clean

# Rebuild from scratch
make clean && make
```

### **Manual Compilation:**

```
gcc -Wall -Wextra -pthread -lrt -o server \
src/main.c src/master.c src/worker.c src/http.c \
src/thread_pool.c src/cache.c src/logger.c \
src/stats.c src/config.c
```

### **Compiler flags:**

- **-pthread:** Thread support
  - **-lrt:** POSIX semaphores and shared memory
- 

## **3. Configuration**

### **Configuration File (server.conf):**

```
PORt=8080
DOCUMENT_ROOT=../www
NUM_WORKERS=4
THREADS_PER_WORKER=10
MAX_QUEUE_SIZE=100
LOG_FILE=access.log
CACHE_SIZE_MB=10
TIMEOUT_SECONDS=30
```

### **Key Parameters**

Parameter	Default	Description
PORT	8080	TCP port for HTTP server
DOCUMENT_ROOT	./www	Root directory for serving files
NUM_WORKERS	4	Number of worker processes
THREADS_PER_WORKER	10	Thread pool size per worker
MAX_QUEUE_SIZE	100	Connection queue capacity
CACHE_SIZE_MB	10	Maximum cache size per worker (MB)
LOG_FILE	access.log	Path to access log file

TIMEOUT_SECONDS	30	Connection timeout in seconds
-----------------	----	-------------------------------

---

## 4. Execution

### Starting the Server:

```
# Compile the code  
make  
  
# Start the Server  
./server
```

### Expected Output:

```
--- Configuration Loaded ---  
Port: 8080  
Workers: 4  
Threads per Worker: 10  
-----  
[MASTER] Logger initialized: access.log  
[IPC] Initializing shared memory and semaphores...  
(...)
```

### Accessing The Server:

```
# Web browser  
firefox http://localhost:8080  
  
# Command line  
curl http://localhost:8080/index.html
```

### Stopping The Server:

```
# From server terminal  
Press Ctrl+C
```

```
# Or send signal  
kill $(pgrep -f "./server")
```

---

## 5. Monitoring

### **Server Statistics:**

Statistics display automatically every 30 seconds

```
=====  
SERVER STATISTICS  
=====  
Uptime: 120 seconds  
Total Requests: 1542  
Successful (200): 1425  
Not Found (404): 107  
Bytes Transferred: 15728640  
Active Connections: 12  
=====
```

### **Access Logs:**

Logs are saved automatically and displayed in the access.log document

### **Log Format:**

```
127.0.0.1 - - [10/Dec/2025:13:55:36 +0000] "GET /index.html HTTP/1.1" 200  
2048
```