

# EN0726 Network Programming

## Sockets for Internet Programming Image

Some of the most exciting software applications today exploit the fact that computers are no longer standalone devices but are able to communicate with each other using computer networks. Web services, file sharing, email, online gaming, database access, home automation, remote monitoring and control etc. depend upon programs that use networks for communication. This module will show you how to develop network programs using sockets. Sockets provide a convenient abstraction of the network so that no matter whether communication occurs over Ethernet, 802.11 wireless or GSM mobile networks, the programmer's view of the network remains essentially the same. The main ideas are presented in a series of lectures. There are ample opportunities for practical work involving the development of network programs. You will study relevant academic and industry-standard literature.

## 1. News

---

Welcome to EN0726 for 2010. Teaching starts on Mon 18-01-2010.

## 2. Module Team

---

### Module Tutor

[David Kendall](mailto:david.kendall@northumbria.ac.uk) david.kendall@northumbria.ac.uk

### Lecturer

*Neil Eliot* neil.eliot@northumbria.ac.uk

## 3. Teaching Arrangements

---

**Lecture** Mon 17.00 - 18.00 EB A102B

**Lab/Seminar** Fri 10.00 - 12.00 EB D003 South

## 4. Synopsis

---

The aim of this module is to provide a grounding in the theoretical principles and practical application of the POSIX socket API.

On completion of this module, students will be able to:

1. Develop an appropriate client/server model for an internet aware application.
2. Critically analyse the sockets interface requirements of a given application.
3. Design, write and test, programs that use the POSIX socket API.
4. Discuss applications, protocols or algorithms of current interest to the network application development research community.

## 5. Teaching Plan

The following is a *provisional* guide to the organisation of the module for this year. These arrangements are subject to change during the course of the module.

Week	W/c	Lecture	Lab/seminar
1	18-Jan	Introduction. [ <a href="#">Slides</a> ] Simple Example [ <a href="#">Slides</a> ]	Admin. Lab familiarisation. Hello World.
2	25-Jan	Socket Preliminaries [ <a href="#">Slides</a> ]	Network Programming Practical [ <a href="#">Lab</a> ] [ <a href="#">Code</a> ]
3	01-Feb	Elementary TCP Sockets [ <a href="#">Slides</a> ]	Network Programming Practical [ <a href="#">Lab</a> ] [ <a href="#">Code</a> ]
4	08-Feb	Elementary UDP Sockets [ <a href="#">Slides</a> ]	Network Programming Practical [ <a href="#">Lab</a> ]
5	15-Feb	Concurrent Servers [ <a href="#">Slides</a> ]	Network Programming Practical [ <a href="#">Lab</a> ]
6	22-Feb	Broadcasting/Multicastin and Socket Options [ <a href="#">Slides</a> ]	Network Programming Practical [ <a href="#">Lab</a> ]
7	01-Mar	Signals and Non-blocking I/O [ <a href="#">Slides</a> ]	Assignment Introduction, Planning and Preparation
8	08-Mar	I/O Multiplexing [ <a href="#">Slides</a> ]	Network Programming Practical
9	15-Mar	Case study	Assignment Reviews
10	22-Mar	Assignment Surgery	Assignment Surgery
11	29-Mar	Assignment Surgery	No lab session. Bank Holiday
12	26-Apr	Assignment Surgery	Assignment Demonstrations

### Note:

In addition to the taught sessions, you are expected to undertake independent and directed learning. On average, you

should be spending about 8 hours per week on this module.

## 6. Assessment

Summative assessment is undertaken via:

1. a group programming project + individual report, assessing all learning outcomes.  
[[Assignment Specification](#) 12th March 2009]

Formative assessment is provided in the form of a variety of small development or analytical exercises with opportunities for discussion and review.

## 7. Recommended Reading

You are strongly advised to obtain a copy of one of the texts below for your personal use. Try the [Northumbria University campus bookshop](#).

- [DON01] Donahoo, M., *TCP/IP Sockets in C: Practical Guide for Programmers*, Morgan Kaufmann, 2001 [[Amazon](#)]
- [GAY00] Gay, W., *Linux Socket Programming By Example*, Que, 2000 [[Amazon](#)]
- [SFR04] Stevens, W.R., Fenner, B., Rudoff, A.M., *Unix Network Programming: Sockets Networking API v. 1*, Addison Wesley, 2004 [[Amazon](#)]

Useful C programming reference books are:

- [HS02] Harbison, S., Steele, G., *C: A Reference Manual*, Prentice Hall, 2002 [[Amazon](#)]
- [KR88] Kernighan, B., Ritchie, D., [The C Programming Language](#), Prentice Hall, 1988 [[Amazon](#)]
- [PC06] Prinz, P., Crawford, T., *C in a Nutshell*, O'Reilly, 2006 [[Amazon](#)]

If you are new to C programming, try the following introduction:

- [KOC04] Kochan, S. *Programming in C*, Sams, 2004 [[Amazon](#)]

The latest edition of a classic text about C programming in a Unix environment is:

- [SR05] Stevens, W.R., Rago, S., *Advanced Programming in a Unix Environment*, Addison Wesley, 2005 [[Amazon](#)]

Selected articles from the technical literature, e.g.

- [IEEE/ACM Transactions on Networking](#)
- IEEE Transactions on Communications
- [IEEE Internet Computing](#)
- RFCs, IEEE & IETF standards
- [IPTPS](#)
- [ACM SIGCOMM](#)

- [ACM/IEEE Symposium on Architectures for Networking and Communications Systems](#)
- Industry white papers and technical documentation
- [PDOS](#)

## 8. Other resources

---

### Standards and references

- [IEEE Std 1003.1-2004](#) -- The POSIX standard
- [C Programming Language standard](#) -- Actually, this is a WG14 working paper for [ISO/IEC 9899:TC2](#), but it reflects the consolidated standard at the time of issue (and it's free!).
- [Standard C](#) -- a handy, online reference to the Standard C language and its library. Now superseded by the references above but still an extremely useful resource.
- [GNU C Library Manual](#) [[Local copy](#)]

### C Programming Tutorials

- [C programming tutorial \(Cardiff\)](#)
- [C programming tutorial \(Strathclyde\)](#)
- [C programming tutorial \(Washington\)](#)

### C Programming Style

- [Rob Pike's Notes on Programming in C](#)
- [AT&T/SUN C Programming Style Guide](#)
- [NASA C Programming Style Guide](#) -- This is 100 pages of good advice for writing solid code. [[Original link](#)]

### Other links

- [RFC Search](#)
- [Beej's Guide to Network Programming](#) [[Local copy](#)]
- [Sockets FAQ](#)
- [Unix/Linux Tutorial for Beginners](#)