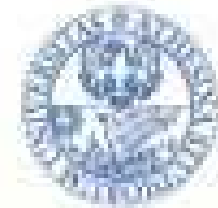


# Reti di Calcolatori AA 2011/2012



UNIVERSITÀ DEGLI STUDI DI TRENTO

<http://disi.unitn.it/locigno/index.php/teaching-duties/computer-networks>

## Programmazione delle socket (lab)

Csaba Kiraly  
Renato Lo Cigno



# Exercises

---

- Boot in Linux
- Tools:
  - text editor
  - gcc
- Exercises:
  - 1, Echo client
  - 2, Echo server
  - 3, HTTP client





# Exercise 1

---

## 1, Echo client:

- Read from STDIN
- Send data to a server, port 5555
- Wait for response
- Write it to STDOUT

## ■ Variants:

1a, TCP, read STDIN and send byte-by-byte

1b, TCP, line buffered (read a whole line, send it)

1c, TCP, new connection for every line

- Example input:

192.168.1.66

Ciao

192.168.1.33

Hi

1d, 1e, 1f,: like above, with UDP





# Excercise 2

---

## 2, Echo server:

- Listen on port 5555
- (Accept connection(s))
- Read data
- Transform data
- and send it back
  
- Variants:
  - 2a, TCP, transform and send back immediately
  - 2b, TCP, line buffered (read a whole line, transform, and send it back)
  - 2c, UDP, read datagram, transform, and send back
  
- Transform:
  - ...





# Exercise 3

---

## 3, HTTP client:

```
httpget <URL> [proxy]
```

- `httpget http://edition.cnn.com:80/WORLD/ proxy.science.unitn.it:3128`

## ■ Tips:

- Parsing proxy: use `sscanf`

```
char hostname[100];  
ret = sscanf(argv[2], "%99[^:]:%d", hostname, &port);
```

- URL parsing (only needed if there is no proxy)

```
char hostname[100];  
int port = 80;  
char page[100];  
int ret;  
ret = sscanf(argv[1], "http://%99[^:]:%99d/%99c", hostname, &port, page);
```





# Cheat sheet

---

```
int sock = socket(domain, type, protocol);
int status = bind(sockid, &addrport, size);
int status = listen(sock, queuelen);
int sock2 = accept(sock1, &name, &namelen);
int status = connect(sock, &name, namelen);
int count = send(sock, &buf, len, flags);
int count = recv(sock, &buf, len, flags);
int count = sendto(sock, &buf, len, flags, &addr, addrlen);
int count = recvfrom(sock, &buf, len, flags, &addr, &addrlen);
int status = close(s);
```

```
struct hostent *he = gethostbyname(const char *name);
```

```
u_long htonl(u_long x); u_short htons(u_short x);
u_long ntohl(u_long x); u_short ntohs(u_short x);
```

```
int status = select(nfds, &readfds, &writefds, &exceptfds, &timeout);
FD_ZERO(&fdvar)      FD_SET(i, &fdvar)      int FD_ISSET(i, &fdvar)
```

