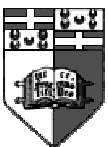


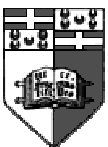
Unix Socket Programming

- BSD sockets provide a way how processes can inter-communicate.
- BSD sockets offer 4 communication protocols which cannot normally be inter-connected: UNIX, Internet, NS, and ISO OSI.
- We will cover only the first two protocols.
- Sockets are **named end point** IPC structure.
- All sockets range over a particular protocol and are of a certain type.
- We will only consider types which provide connection-oriented, reliable, sequenced and unduplicated flow of data, known as stream sockets.



Unix Internet Preliminaries

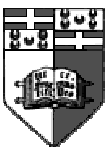
- Our discussion will only skim the surface of this vast subject. A more in depth review will be done in ‘Distributed and Parallel Systems’.
- Every computer on the internet has associated to it a unique identifier which is used to pass data to and fro.
- This identifier can be seen as the postal address of each computer (host). Computers in between two end points act as carriers to these messages.
- Delivery of each message is the responsibility of these intermediate computers and no central forwarding authority exists.
- This addressing protocol is called the IP protocol.





IP and DNS

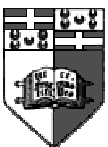
- These identifiers are called IP addresses and are 32 bit numbers.
- IP addresses are normally displayed as dot separated values ex.193.188.34.119 for *babe*
127.0.0.1 is special for *loopback*
- For each message that goes through the internet, the source and destination address must be specified.
- A specific protocol exists which allows human readable values to be translated into IP addresses. This is called the Domain Name System (DNS).
- A hierarchy exists giving shared responsibility to resolve names to IP addresses for all names in the world.
- A typical name would be *babe.cs.um.edu.mt* which translates to *babe's* IP address using the DNS.





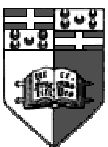
IP and TCP/IP

- IP provides the way how to find a host to deliver messages to, yet it does not guarantee reliable flow of data.
- On the network, errors occur and messages are sometimes lost or corrupted.
- Also the order messages arrive is not guaranteed.
- The **Transmission Control Protocol** resides on top of IP and guarantees data flow reliability and flow control. Also it makes connection-oriented transmission possible.
- For a connection on a host (single IP address), there exist many entry points through which there may be many-to-many connections. These are called **ports**.



Unix IP and TCP/IP cont.

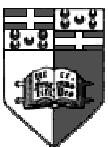
- Ports can only sustain many-to-one connections on the server side and they are numbered from 1 to 65535.
- Internet ports 1-1023 are considered reserved and in fact can only be used by the superuser.
- Ports from 1024 to 5000 can be used at will and will also be assigned automatically by the system.
- The command *netstat* tells you what connections are open at any one time on the system. *netstat* reports all connections including UNIX sockets, etc.
- When a connection is dropped on both ends, the system cleans up any pending connections.





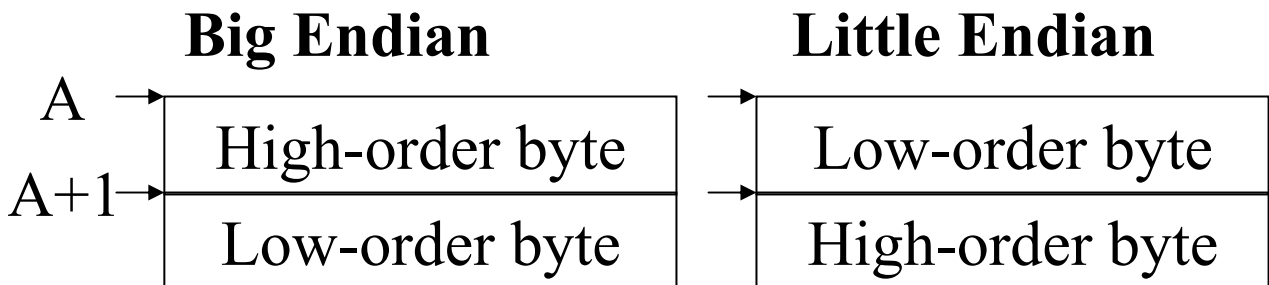
TCP/IP sockets

- Sockets are able to open a connection on a port and transmit whatever data, to and from the recipient.
- To establish a connection through a port, the following tuple must be totally defined in the system:
$$\langle \text{protocol, local-addr, local-port,} \\ \text{foreign-addr, foreign-port} \rangle$$
- In a server-client setup, the server provides the *local* attributes and then waits for a connection from the client. A connection from the client provides all the necessary data to fill this tuple.
- The client initiates the connection to the remote server, using a known remote IP address and port number on which the server is listening. This also provides all the data for the *local* part of the tuple.
- On the whole internet, each of these tuples must be unique.

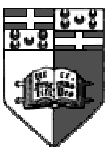


Unix Network Byte Ordering

- Some computer architectures are big endian and some are little endian.
- Intel is actually little endian while the Motorola 68000 is big endian.



- To be able to communicate on the internet, a network byte ordering has been defined.
- Network byte ordering is big endian for 16 and 32 bit integers.
- When passing values that need to be used at the network layer, we need to make the appropriate conversions.

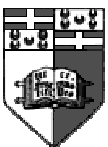


Unix Network Byte Ordering (cont)

- The following library functions handle the **potential** byte order differences between different computer architectures.
- ‘h’ stands for host values while ‘n’ stand for network values.

```
#include <sys/types.h>
#include <netinet/in/h>

u_long htonl(u_long hostlong);
u_short htons(u_short hostshort);
u_long ntohl(u_long netlong);
u_short ntohs(u_short netshort);
```

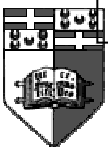




Byte Operations

- Whenever a series of bytes have to be copied, use *bcopy*.
- *bcopy* is better suited than *strcpy* since a series of bytes might contain ‘/0’ inside it.
- When using network structures, be sure to apply *bzero* before using them.
- *bcmp* returns 0 if the byte strings are equal, non-zero if not.

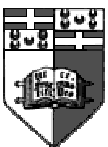
```
#include <string.h>
void bzero(char *src, char*dest, int nbytes);
void bcopy(char *dest, int nbytes);
int bcmp(char *ptr1, char *ptr2, int nbytes);
see above for return value
```



Unix Address Conversion

- Given an IP address (for example 193.188.34.81) stored inside a string, *inet_addr* returns the address in network byte order.
- *inet_ntoa* performs the opposite operation.
- Every subsequent call to *inet_ntoa* overwrites the statically returned string.

```
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
unsigned long inet_addr(const char *ptr);
char *inet_ntoa(struct in_addr inaddr);
```





Exercises

- Find out if babe is big or little endian.
- Find out if your machine at home is big or little endian if you have a UNIX system installed.
- Try using the *ping* application to find out if there is anyone at the other end of the IP addresses you give it (`‘/usr/sbin/ping IP’`).
- Using the ping application, find out the IP address of the DNS name you give it (-s option).

