Computer Systems

Programming: The whole story

Dr. Mario Gongora
mgongora@dmu.ac.uk

Contents

- Program Organisation
- Functions and Libraries
- Interrupts and other I/O techniques
- Communications
- Data processing

Program Organisation

- For a program to work it need always:
  - Initialise Memory
  - Initialise Peripherals
  - Load the necessary functions (libraries)
  - Load the necessary I/O service routines (drivers)
  - Wait/load data
  - Process.

Program Organisation

- These elements can be sorted in varied ways:
  - Initialise Memory and Peripherals: **Operating system, Firmware, programmer**
  - Load the necessary functions: **Compiler, programmer**
  - Load the necessary I/O service routines: **O/S, compiler or programmer**
  - Wait/load data: **programmer**
  - Process: **programmer**

Functions and Libraries

- It would be inefficient to write always everything from scratch
- The programmer can reutilise parts his/hers own programs (either within a same program or in various projects): **Functions**
- We can also use parts or programs written by others (e.g. Microsoft, the developers of the Compiler or the manufacturer of a piece of hardware): **Libraries**

Interrupts and other I/O techniques

- When programming, an action plan for attending the I/O has to be devised.
- There are various types of I/O needs:
  - Asynchronous I/O (e.g. Keyboard, mouse, alarms from timers, etc…)
  - Synchronous or periodic I/O (e.g. Temperature sensor)
  - Bulk or Massive Data I/O (e.g. Hard Disk)

Interrupts and other I/O techniques

- It is not possible to predict when an Asynchronous I/O device needs attention.
- Interrupts can be used to service the device when I/O data is available:
  - It does not need permanent attention
  - Needs hardware capability to interrupt
  - Needs to be prioritised and latency evaluated
  - Need to have a stack to save the program's state when the interrupt occurs and a service routine

Interrupts and other I/O techniques

- Synchronous or periodic I/O can be scheduled for attention at regular pre-established intervals.
- Polling can be used to service these devices:
  - Latency and priorities can be pre-established
  - No specialised hardware needed
  - The program can be organised to attend in non-busy periods

Interrupts and other I/O techniques

- Is too inefficient to service Bulk or Massive Data I/O by means of the CPU (slow to access).
- DMA can be used to service these devices:
  - The peripheral can be given direct access to the memory and let to transfer the data itself
  - The CPU can work in other processes meanwhile
  - Special hardware is needed

Communications

- The program can communicate with other programs in the same or other machines
  - Inter-Process communications
    - Memory Stacks, Sockets, Semaphores, etc.
  - Serial ports (point to point)
    - RS232, etc.
  - Networks
    - CAN, TCP/IP, Bluetooth, etc.

Data processing

- The main function of a program.
- Once data has been collected, communications achieved, etc…, the processing is performed
- The result is presented in various ways:
  - Data saved in storage (files)
  - Actions taken with peripherals (actuators)
  - Data sent elsewhere (networks)