2 Background

Read this and the next section **before** *you attend the first class.*

2.1 Basic Structure of a Computer

Before embarking on computer programming it is useful to have some idea about what a computer is, what it can, and more importantly, cannot do. The basic structure of a digital computer is outlined in figure 1.

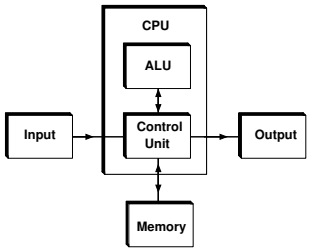

Figure 1: Basic structure of a digital computer

The *Central Processor Unit* (CPU) operates on a series of instructions called a *program* which is stored in *memory*. The CPU consists of an *Arithmetic Logic Unit* (ALU) which performs such functions as addition, subtraction, comparison etc. and a *Control Unit* which directs and monitors the operation of the computer. The *input* and *output* provide an interface to the outside world, for example *inputs* can be the keyboard and *output* the display screen. The *input* and *output* units also provide access to data storage devices such as hard discs, floppies, CD-ROMs etc. All these may be either connected directly to the computer or over the *network*.

The operation of the computer is based on the *Fetch - Execute* cycle as shown in figure 2 where the *Control Unit* continually FETCHES instructions and data from memory and EXECUTES them using the ALU. This cycle continues the whole time the computer is switched on.

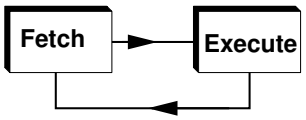

Figure 2: Fetch and Execute cycle

The computer holds all information (number, letter, instructions), in memory as patterns of binary bits, (ones and zeros). The instructions are decoded by the *Control Unit* and determine what operations the ALU performs on the data. The desired sequence of instructions (the PROGRAM) have to be placed in memory by the "programmer", which is what we will be doing during this course. Remember that the computer is a machine that exactly follows instructions and it is the sole responsibility of the programmer to get these instructions right!

The binary instructions used by the computer are not convenient for humans to use. This is further complicated by the fact that different computers use different sets of instructions and even different format for numbers. It it therefore much more convenient to program in a HIGH

LEVEL LANGUAGE which is (more) humanly readable and, to a large extent, independent of the details of the computer being used. The high level language chosen for this course is JAVA.

2.2 The Computers being Used

This course will be run on LINUX systems from within the COMPUTATIONAL PHYSICS LABORATORY or via for remote connection from the MICROLABS to a compute server. The actual system is connected up as shown in figure 3

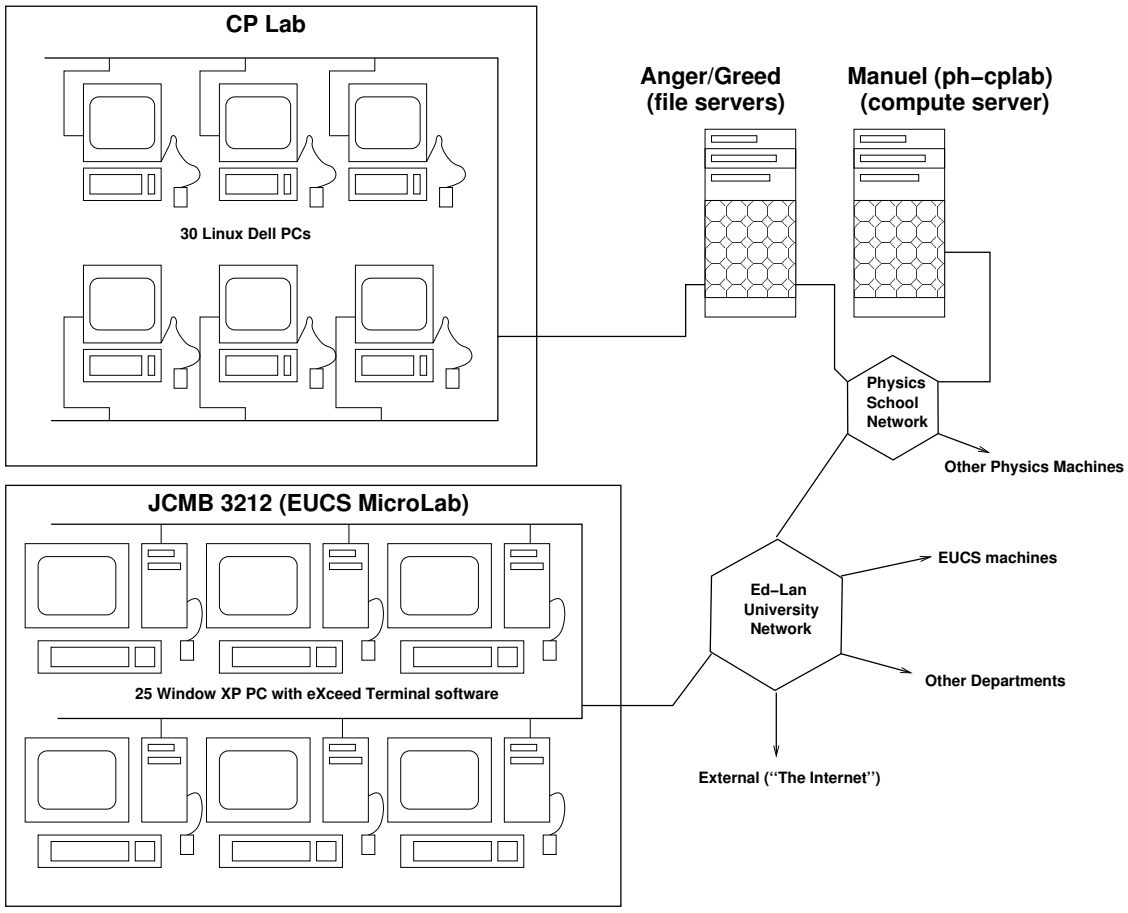

Figure 3: Layout of Computational Physics Laboratory

All users will see the same LINUX envorinment whether seated at a computer in the COMPUTATIONAL PHSYICS LABORATORY, or in the EUCS MICROLAB[1]. All user files and data are held on the fileserver so you can work from any of these computers. It is also possible to connect to the Physics server from other University microlabs, for example MAIN LIBRARY.

2.3 The UNIX operating system

The "*operating system*" controls the computer hardware, allows programs to run and controls the users interface. The UNIX operating system with X-WINDOWS interface is widely used in

[1]Once connected to the server.

the scientific and computer science areas. It is a "multi-user" system where, unlike PCs, more than one person can use the machine at once without "seeing" each other. UNIX is used on medium sized workstations and servers right up to massive super computers.

The user interface is via the X-windows system which is "windows and mouse" based, but most interaction with the system is via a "TERMINAL WINDOW" into which you type commands. In addition the different utilities are less well integrated than on the PC, although things are slowly getting better.

For PC users the "command" interface initially appears rather old-fashioned but with a bit of practice it becomes very natural and in many ways easier and faster that the multi-menu, sub-menus, sub-sub-menus, pop-up menus, pop-down menu, tear-off menu, found on the MS-Windows and the Office utilities.

2.4 Components of a program

A computer program written in a *High Level Language* has to be converted into binary instructions that are executed by the processor. Using JAVA this is a two stage process, the components of which are show in figure 4

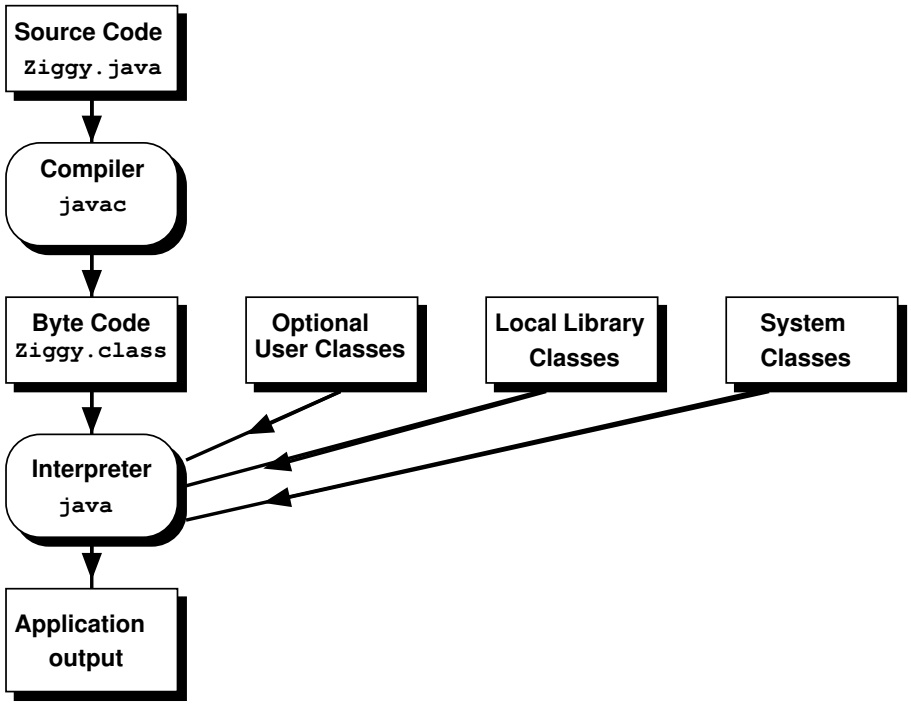

Figure 4: Components of a JAVA computer program

1. The actual JAVA program is held in a file on disc which is called the *source code*. You create the file with an *editor*. This file can also be viewed on the screen or printed.
2. This *source code* is COMPILED to produce *byte code*. This is done with a COMPILER.
3. The *byte code* [2] is then combined, (or LINKED) with other *byte code* files and library files and executed by the *Interpreter*, hence running the program.

[2] JAVA differs from most other languages in that the output form compiler is independent of the machine type.

Utilities Used in Program Writing and Preparation.

The three utilities you will use in this course are:

1. **Editor:** This allows to to write, edit and store on disc the *source code* of the program. Most large computer systems offer you a choice of editors, but they all basically to the same thing. The editor being used in this course is `emacs`. See section on the use of `emacs`.

2. **Compiler:** This *compiles* the *source code* into *byte code*, this is called `javac`.

3. **Interpreter:** This executes the *byte code* on the particular machine after linking in the relevant additional "classes". This is called `java`.

These are the three basic "tools" used to write programs and the only two that will be needed in this course.