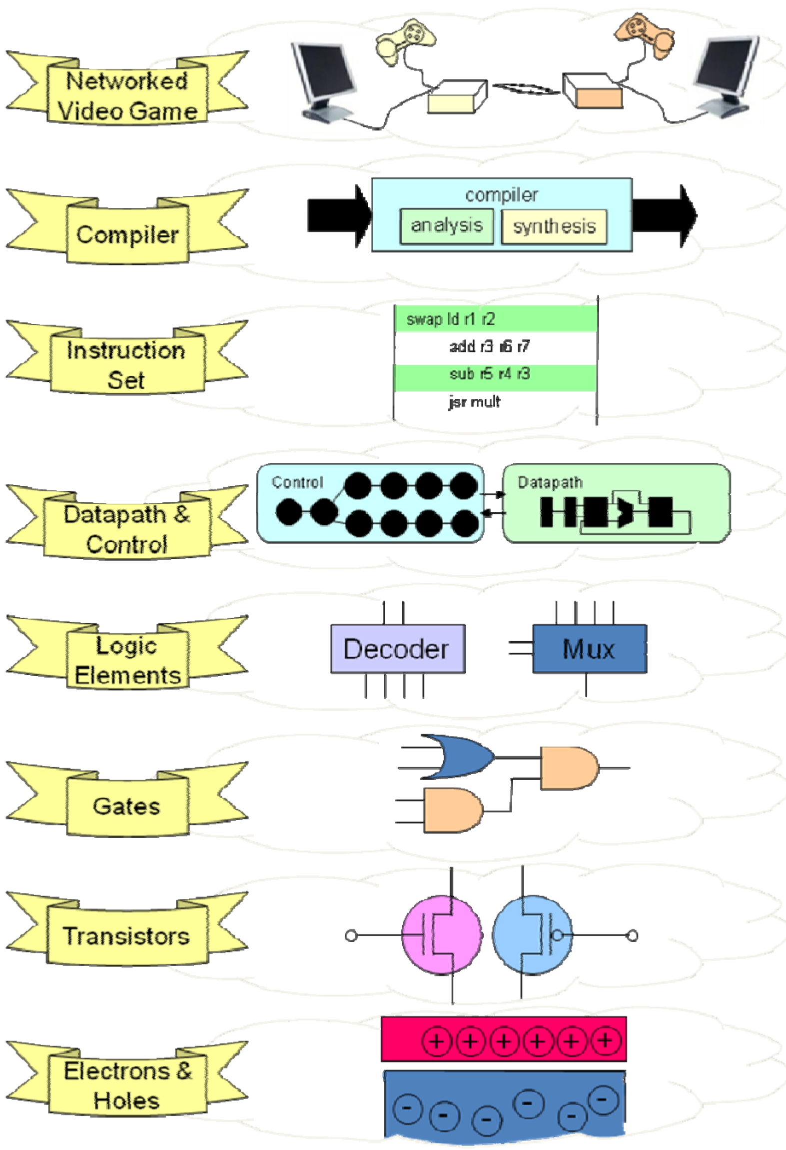

Figure 1.5: From Electrons and Holes to a Multi-player Video Game

1.3 The Role of the Operating System

Where does the operating system figure in the above discussion? It is the resource manager that is responsible for orchestrating the use of the hardware resources for the whole endeavor right from the design of the game to the actual game playing. To see this, let us return to the video game example. We have written the client-server application in high-level language. We may use a simple text editor or a sophisticated program development system such as Visual Studio for the development of our video game. Once developed, we compile the program to the instruction-set of the processor. Each of the text editor, the compiler, etc., is a program that needs to run on the processor. For example, the compiler has to run on the processor taking the HLL program as input and produce machine code as the output. The operating system makes the processor available for each of these programs to its job. Now let us see what happens when you are actually playing the game.

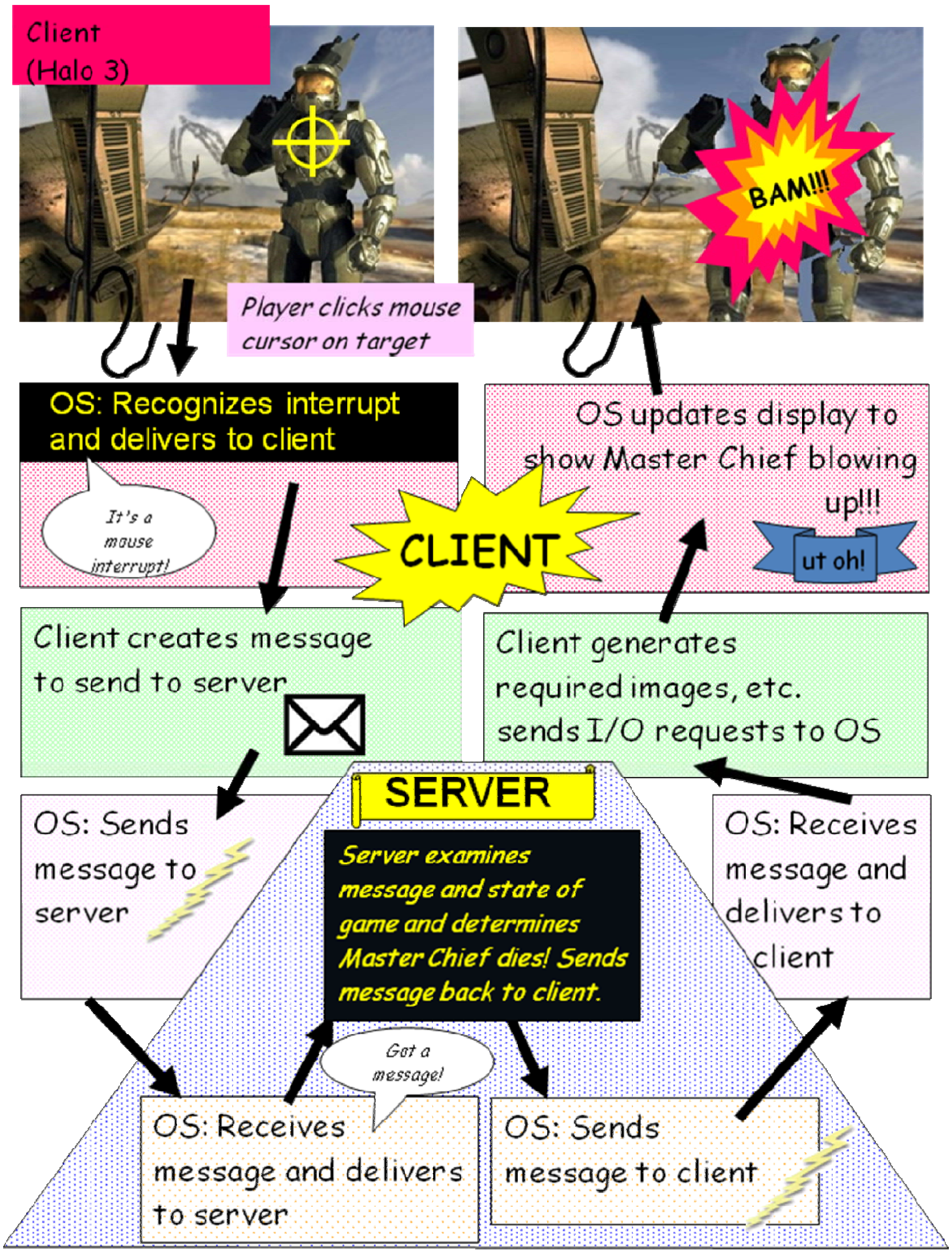

Figure 1.6: Application-Hardware-OS interactions

In the video game example, you click the mouse button that results in the master chief blowing up on your screen and the screens of every one of the players (see Figure 1.6).

What is going on? First, the hardware device controller on your box records your mouse click. The controller then *interrupts* the processor. Remember that the processor is currently running your client program. An interrupt is a hardware mechanism for alerting the processor that something external to the program is happening that requires the processor's attention. It is like a doorbell in a house. Someone has to see who is at the door and what he or she wants. The operating system (which is also a collection of programs) schedules itself to run on the processor so that it can answer this doorbell. The operating system fields this interrupt, recognizes that it is from the mouse intended for the client program, and hence passes it on to the client program. The client program packages this interrupt as a message to the server program through the network. The server program processes this message, updates the state of the game incorporating the new information, and sends messages to all the clients with the updated state. The clients, through their respective operating systems, update their respective displays to show the new world state. As can be seen several hardware resources (processor, memory for the programs and data, mouse, display, network connections) are all being allocated and de-allocated in the course of clicking the mouse button to the point where the display changes. The operating system orchestrates all of these actions.

1.4 What is happening inside the box?

The video game example serves as a teaser for the interactions between applications, operating system, and hardware. To get a good understanding of what is going on inside the box we have to get a good handle on what is happening inside the box with both the system software and the hardware architecture.

Figure 1.7: From PDAs to Supercomputers

First, it is useful to understand that there are several instantiations of computer systems ranging from handheld devices such as cellphones and PDAs, tablet PCs, notebook PCs, desktop computers, parallel machines, cluster computers, to supercomputers, as shown in Figure 1.7.

Interestingly, irrespective of these different manifestations, the organization of the hardware inside a computer system is pretty much the same. There are one or more central processing units (CPUs), memory, and input/output devices. Conduits for connecting these units together are called buses, and the device controllers act as the intermediary between the CPU and the respective devices. The specifics of the computational power, memory capacity, and the number and types of I/O devices may change from one manifestation of the computer system to the next. For example, commensurate with its intended use, a PDA may have very limited I/O capabilities such as a touch-screen display, microphone, and speakers. A high-end supercomputer used for running large-scale scientific applications such as modeling climate changes may employ several hundreds of CPUs, incorporate several Gigabytes of memory, and be connected to an array of disks with storage capacity on the order of several Terabytes. Figure 1.8 shows the organization of the hardware in a typical desktop computer system.

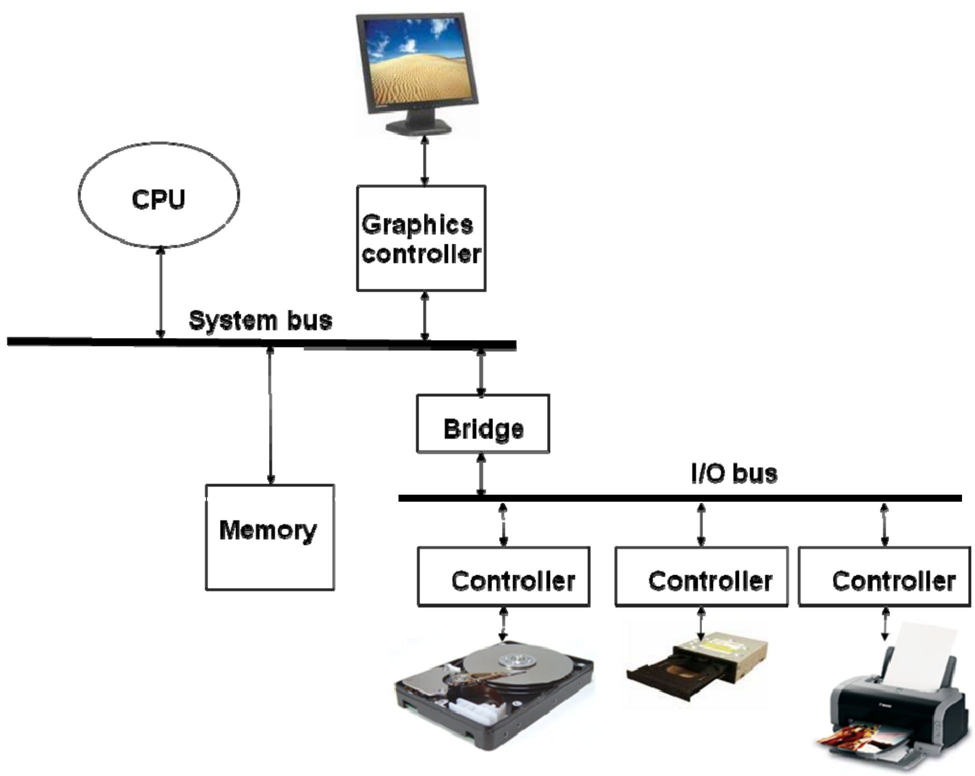

Figure 1.8: Organization of Hardware in a Desktop Computer System

The organization suggests that there is scope for simultaneous activity of the hardware elements (i.e., concurrency). For example, it should be possible for the printer to be printing some document while the hard drive is reading an MP3 file from the disk to play your desired music while you are reading some news story from CNN through your web browser. The CPU is the brain of the entire system. Everything that happens in the computer system is a result of some program that runs on the CPU. You may observe that simultaneous with your watching CNN on your computer screen the printer may be busy printing some document for you from a document-editing program. Web browser is an application program and so is the document editor. The operating system gives time on the CPU for each of these programs to initiate their activities so that the hardware concurrency that is suggested by the organization is actually realized in practice.

1.4.1 Launching an application on the computer

Let us understand how the various entities shown in Figure 1.8 working with the operating system come together to give you a simple computing experience, say of watching a video on the display device. The box labeled "memory" holds any program that you wish to execute on the CPU. In the absence of any user program, the operating system, which is also a program, is always executing on the CPU, ready for work that the user may want the computer system to carry out. First, with your mouse you may click on an icon on the display device that says "movie player". The movement of the mouse and the mouse click are fielded by the operating system. From the icon clicked, the operating system knows which program the user wishes to execute. All such programs are resident on the hard drive of the computer. The operating system "loads" the executable image of the movie player program into the memory and transfers control of the CPU to start executing this program. The execution of the movie player program results in opening a graphics window on your display and asks the user to specify the specific movie file that the user wishes to watch. You would then use a keyboard perhaps to type the name of the file including the drive on which the file may be found (let us say, the DVD drive). The program then opens the file on the DVD drive and plays it, and you are now ready to watch your favorite movie on the display device. The operating system is involved in every step of the way to deliver you the movie watching experience including: (a) updating the graphics display, (b) capturing the user inputs on the keyboard and delivering them to the movie player program, and (c) moving the data from a storage device such as the DVD drive to memory. The actual mechanics of data movement from/to the I/O devices to/from the memory and/or the CPU may depend on the speed characteristics of the devices. We will discuss these aspects in much more detail in the chapter on I/O subsystem (Chapter 10). The I/O bus and system bus shown in Figure 1.8 serve as the conduits using which data is moved between the various sources and destinations depicted by the hardware elements. Just as highways and surface streets may have different speed limits, the buses may have different speed characteristics for transporting data. The box labeled "bridge" in Figure 1.8 serves to smooth the speed differences between the different conduits in use inside a computer system organization.

1.5 Evolution of Computer Hardware

With the ubiquity of computing in our everyday living, it is difficult to imagine a time when computers were not so commonplace. Yet not so long ago, the computational power in your current day laptop that you may have purchased for less than $1000 cost over a million dollars and occupied a room the size of a large dance floor with elaborate cooling and raised floors for the cabling.

In the early 40's, ENIAC – Electronic Numerical Integrator and Computer – was built at the University of Pennsylvania and is largely considered as the very first programmable electronic digital computer (Figure 1.9).

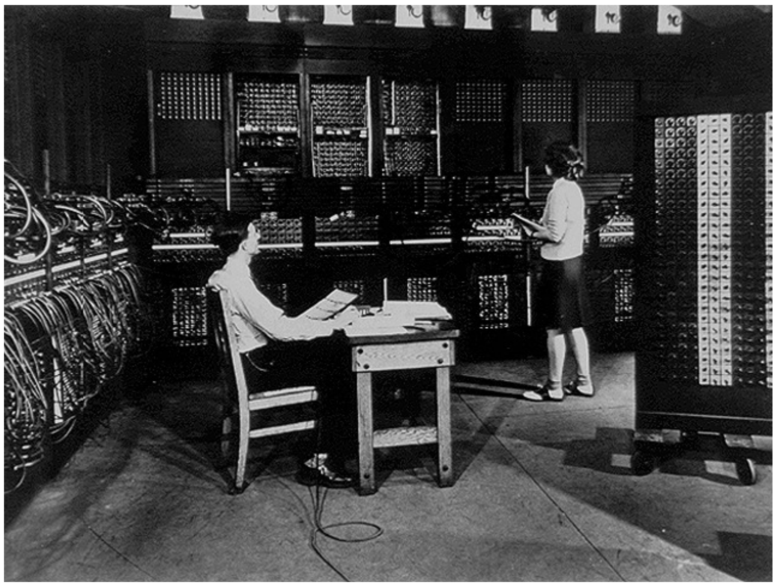

Figure 1.9: ENIAC, first electronic digital computer

Built with over 18,000 vacuum tubes, 1000 bits of random access memory implemented using tiny magnetic ferrite cores, and consuming around 170 Kilowatts of electrical power, the ENIAC had a processing power that is roughly equal to what you may find in a modern day musical greeting card! This is how far computing technology has progressed in a little over 6 decades since the time of ENIAC.

One can attribute the rapid progress of computer hardware due to the ingenuity of scientists and engineers in complementary yet diverse fields including physics, chemistry, electrical engineering, mathematics, and computer science. Of course, the most visible technological revolution that has spurred the rapid advance of the computer industry is in the semiconductor revolution. Digital computers had their modest beginnings using vacuum tubes and magnetic core memory in the 1940's. With the invention of a switching device called the "transistor" at Bell Labs in 1947, the semiconductor revolution began to take shape. The era of digital computers built with discrete transistors gave way to the integration of several such transistors on to a single piece of silicon. Microchip – single chip microprocessors based on CMOS transistors employing Very Large Scale Integration (VLSI) – introduced in the late 80's and early 90's, was perhaps the tipping point in the computer hardware revolution. Today every computing device from cell phones to supercomputers uses the microchip as the basic building