

CS101 Introduction to Computing

Lecture 10

Computer Software



Lecture 8 was on the binary number system and logic operations

1. About the **binary number system**, and how it **differs** from the decimal system
2. **Positional notation** for representing binary and decimal numbers
3. A **process** (or algorithm) which can be used to **convert** decimal numbers to binary numbers
4. Basic **logic operations** for Boolean variables, i.e. NOT, OR, AND, XOR, NOR, NAND, XNOR
5. Construction of **truth tables** (How many rows?)



Learning Goals for Today

1. To discuss the **role of software** in computing systems
2. To **learn to differentiate** among software belonging to the system and application categories
3. To learn about software **ownership**



We mentioned in Lecture 4 that at the highest level, **two things** are required for computing

Hardware: The **physical equipment** in a **computing environment** such as the computer and its **peripheral devices** (printers, speakers...)

Software: The **set of instructions** that **operates** various parts of the hardware. Also termed as “computer program”



Computer Software

- The **HW needs SW to be useful**; the SW needs HW to be useful
- When the user needs **something done by the computer**, he/she gives instructions in the form of SW to computer HW
- These **instructions need to be written** in a language that is readily **understood** by computer **uP**



Machine Language

- That language is called the machine language
- Machine language, though readily understood by microprocessors, is very difficult to write in for human programmers
- Language translators were invented to overcome this problem



Language Translators

- **Human** programmers **write programs** in a language that is **easy to understand for them**
- They use language translators to **convert** that program into machine language – a language that is easy to understand for the uPs
- We'll have more to say about the machine language and language translators in a **future lecture**

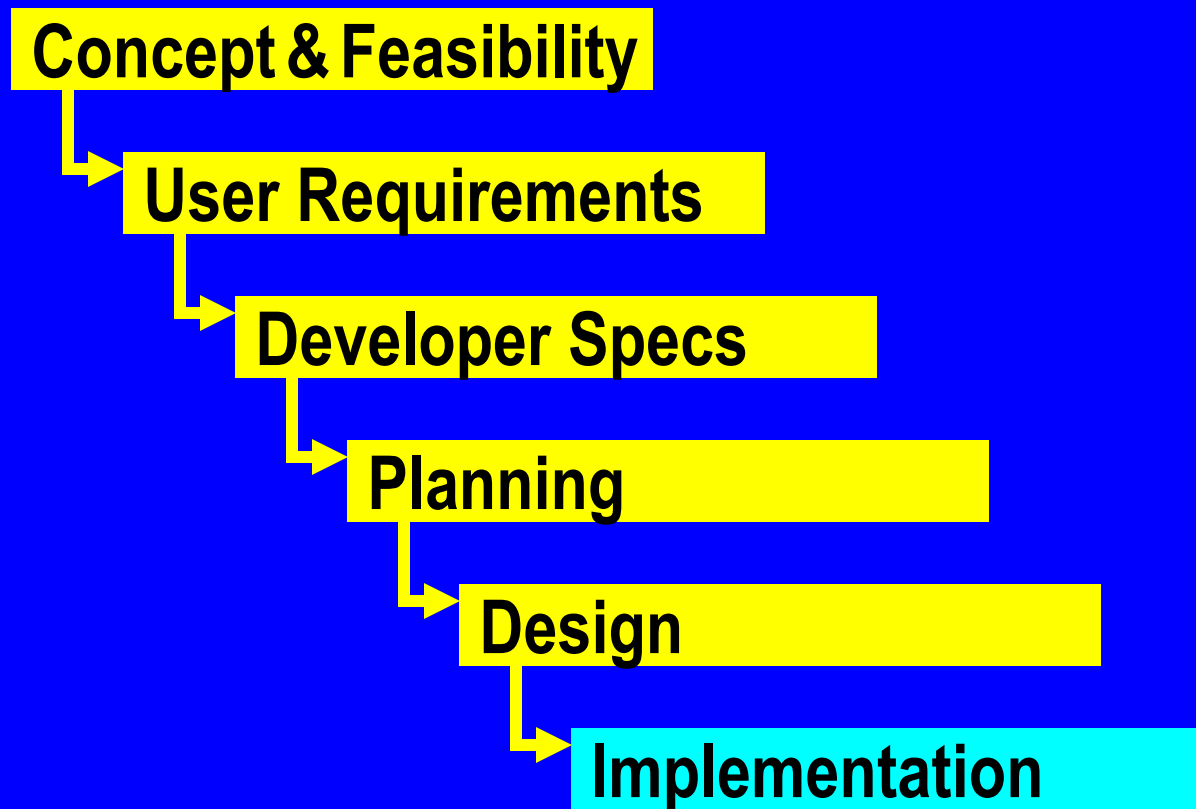


Software Development

- Writing very **short programs is easy**, but developing **reasonably-sized programs** is quite difficult as you are going to learn over the **next few years**
- The SW development process involves **many steps**, and **coding**, that is typing the instructions in a high-level language is only a **small part of that process** – taking-up only around **15% of the effort**
- A **summary** of the steps involved is shown on the **screen**. We'll have more to say about them during the **20th lecture**



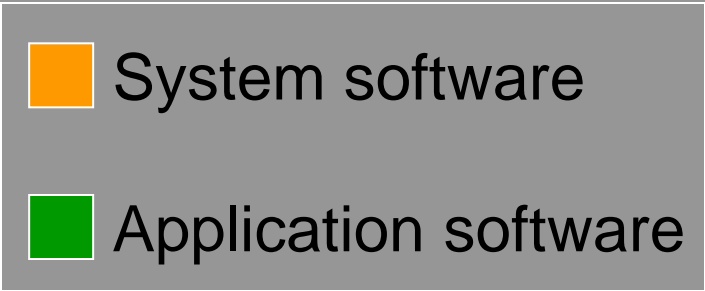
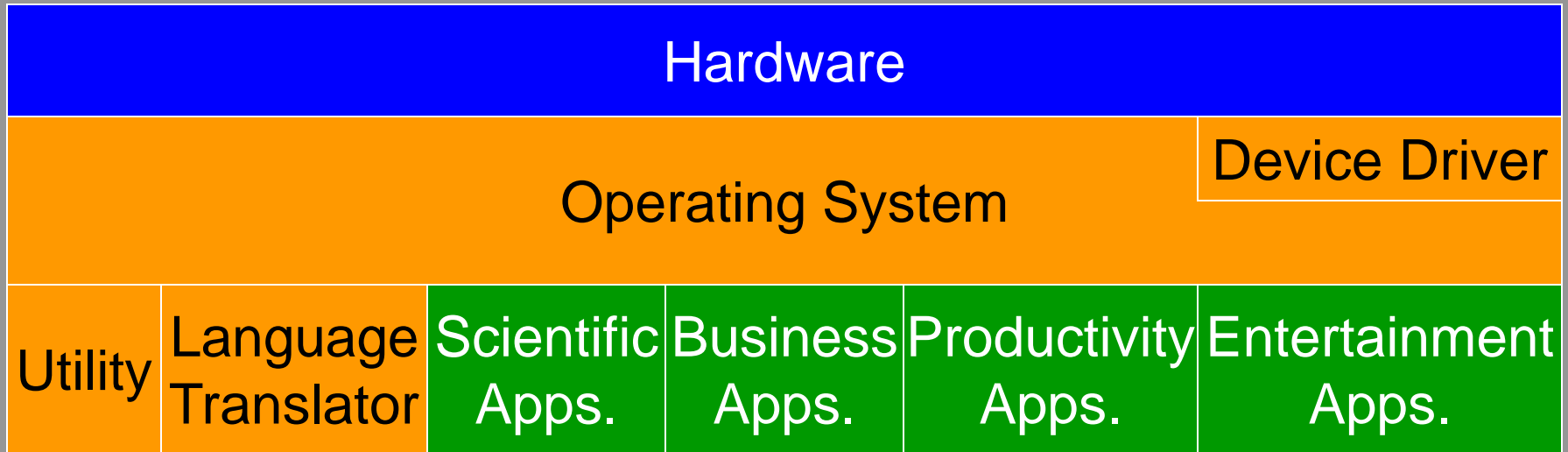
The Software Development Process



Two Major Types of SW

- **System SW**
 - Programs that generally perform the **background tasks** in a computer. These programs, many times, **talk directly** to the HW
- **Application SW**
 - Programs that generally **interact with the user to perform work** that is useful to the user. These programs generally talk to the HW through the **assistance of system SW**
- The **diagram** on the screen shows the **relationship** between HW and these two types of SW





System SW are programs that ...

- Control the overall operation of the computer
 - OS
- Interact directly with HW
 - Device drivers
- Perform system management & maintenance
 - Utilities
- Are used to develop or maintain other programs
 - Language translators



Operating System

- Performs its work invisibly to **control the internal functions of a computer**, e.g. maintaining **files** on the disk drive, managing the **screen**, controlling which **tasks the uP performs** and in what **order**
- It **interacts directly** with the computer HW
- **Other SW** normally does not directly interact with the HW, but **through the OS**
- **Examples:**

Windows

Mac OS

Linux

Unix

Solaris

DOS

CP/M

VMS



Firmware

- OS components that are stored permanently on chip (ROM) and not on the disk drive
- When a computer is powered-on, firmware is the first program that it always executes
- Firmware consists of startup and a few low-level I/O routines that assist the computer in finding out and executing the rest of the OS
- On IBM-compatible PC's, it is called BIOS



Utilities

Computer programs that perform a **particular function** related to computer **system management and maintenance**

Examples:

1. **Anti-virus** SW
2. Data **compression** SW
3. Disk **optimization** SW
4. Disk **backup** SW



Language Translators

Programs that take code written in a HLL and translate it into a low-level language that is easily understood by the uP

1. **Compiler** translates the program written in a HLL **in one go**. The translated code is then used by the uP whenever the program needs to be run

2. **Interpreter** translates the HLL program **one statement at time**. It reads a single statement, translates it into machine language and passes that machine language code to the uP and then translates the **next statement, and so on ...**



Device Drivers

- A computer program that **facilitates the communication** between the computer and a **peripheral device** (e.g. printer, mouse, etc.)
- It takes the instructions and/or data from the computer and **converts them into a form** that is readily understood by a peripheral device, **and vice versa**

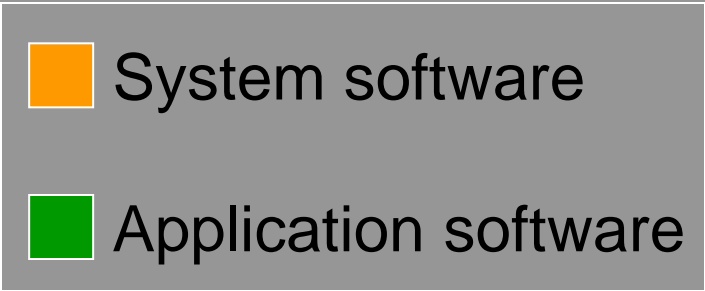
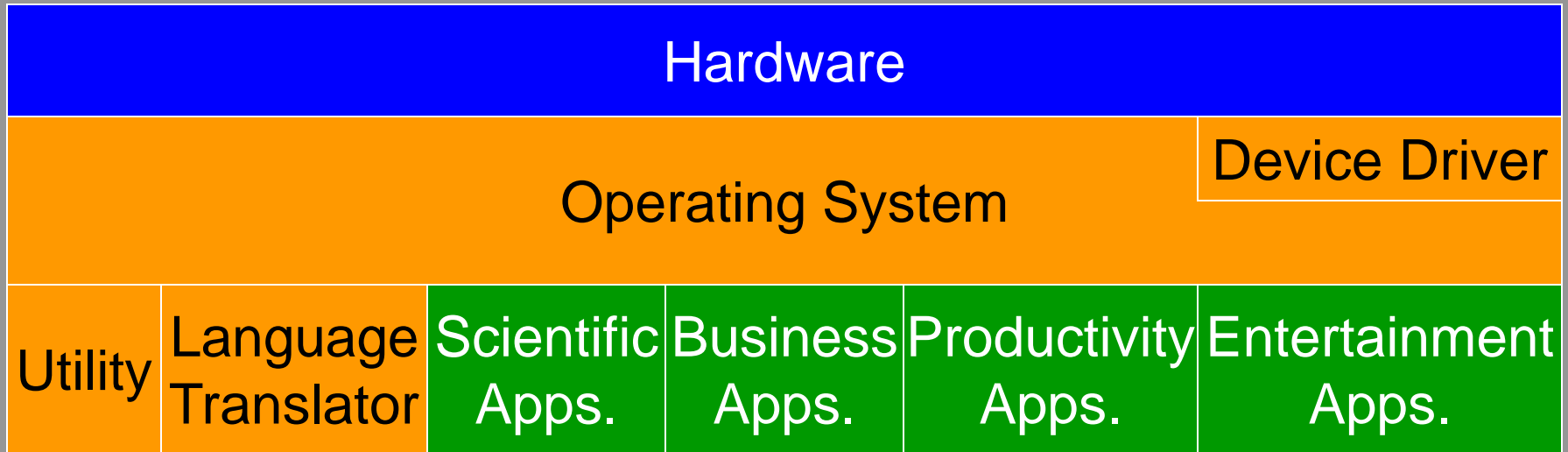


Application SW

Application SW are programs that **interact directly with the user** for the performance of a certain type of work

- **Scientific/engineering/graphics SW**
 - Mathematica; AutoCad; Corel Draw
- **Business SW**
 - The billing system for the mobile phone company
- **Productivity SW**
 - Word processors; Spreadsheets
- **Entertainment SW**
 - Games
- **Educational SW**
 - Electronic encyclopedias; The VU Web site





Another way of classifying SW

- **Shrink-Wrapped SW**
 - You can just **go to a shop** and buy it
- **Custom-built SW**
 - You cannot just go to a shop and buy it; you have to **find someone who can develop** it for you



Shrink-Wrapped SW

- SW built in such a way that it is **useful for many different users in many different ways**
- Example: MS Word. **Individuals** use it and so do many **large corporations**. It is used for writing one-page **letters** and also to typeset **books**



Custom-Built SW (1)

- SW built for a particular organization to fulfill the needs of that particular organization
- **Example:** A system for predicting the preferences of the Northwest Airline pilots
- This type of SW is **expensive** because the builder has to recoup costs and make a profit from a single sale



Custom-Built SW (2)

- The **delivery time** is longer
- Customers get **more productivity** out of it because it is built according to their **exact specifications** – just like a **custom-built shoe fits better**, but generally is more **expensive**, and requires a **longer period for delivery**



Who Owns Software?

- Generally, although a piece of SW that is being used by millions, it is **not owned by any of them!** Instead, it is owned by the maker of the SW
- The makers let us use their SW but keep the ownership to themselves. When we buy a SW package, we do not really buy it – **we just buy a license** that allows us to use it, the ownership stays with the maker
- However, there are **variations** on this theme



3 main types of SW licensees

1. **Proprietary** – Most software on a Windows PC or a Macintosh belongs to this category
2. **Freeware** – Most software on a Linux PC belongs to that category
3. **Shareware** – the category which lies between the above two categories



Proprietary SW License

- The user **needs to pay the maker of the SW for buying a license** that allows the user to use the SW
- The license, generally, **does not transfer the ownership** of the SW; it just allows the user to use it
- The user is **legally barred from making copies** of the licensed SW. Generally, the license is for the **personal use only**
- **Most SW in use** in the world is of this type
- **Examples:** Windows, Mac OS, MS Word, Adobe Photoshop, Norton Antivirus



Types of Proprietary Licenses

- Single-user license
- Multi-user license
- Concurrent-user license
- Site license



Freeware SW License

- Also known as “Public Domain SW”
- Allows the user free use of the SW
- The author, however, generally retains ownership
- Can usually be downloaded from various Web sites
- Examples: Linux; LaTeX; Netscape Web browser – the Navigator; MS Web browser – the Internet Explorer
- Why give away SW for free? (message board)



Open-Source SW License

- Some authors give away the **machine code** only, which is extremely **difficult to modify**, if at all!
- Others even give away the **high-level language source code** so that users can **make changes** according to their own requirements
- The later practice is called **open-source licensing**
- Examples: **Linux**; Netscape **Navigator**



Shareware SW License

- Allows the user **free use** of the SW, but with **a request** that the user pay the author a small amount (US\$10-50) if the user is satisfied with the SW
- The **author retains ownership**
- Can usually be **downloaded** from various Web sites
- Examples: **WinZip**, **Download Accelerator**
- Why give away SW (initially) for free?
 - The author is an individual or a small business that **cannot afford to advertise**. No one'll even try the SW if it had a price
 - The **expectation** is that the user will try the SW for free, **find it useful**, and then pay the very small price for the SW

Trialware

- **Similar** to shareware, **but different**
- The SW is **usable for a short period only**
- After an initial trial period that can range from a week to a few months, the SW **self-destructs**
- Can be **downloaded** from the Internet or alternatively, the user can receive a copy by snail-mail by writing to the maker of the SW
- **Why trialware?**
 - So that the customer can have a **risk-free trial** for a limited-period only



What have we learnt today?

1. We have found out about the **role** software plays in a computing environment
2. We also learned to **distinguish** between software belonging to the **system** and **application** categories
3. We also discussed the different types of software **licenses**



Topics of some of the future lectures

- Operating system
- Application SW
 - Productivity SW
 - Word processor
 - Spreadsheets
 - Presentation making
 - Databases
- Programming Languages
- The SW development process
- The Web development series of lectures is clearly focused on developing SW



Focus of the Next Lecture

- The **role** of the **OS** in a computing environment
- The various **functions** that an OS performs
- The main **components** of an OS
- Various **types** of OSes

