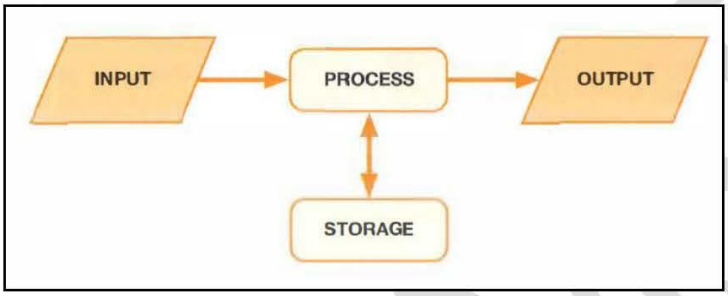
Chapter 16

Input Process Output

input to **enter** data into a computer

process to **change** the meaning or format of some data

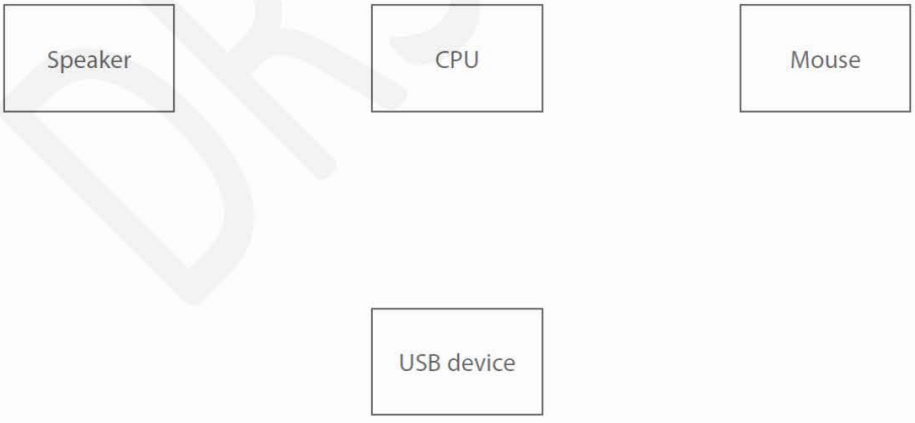
output to **display** or output data that has been processed



(b) Computers are made up of hardware components.

(i) Complete the diagram by adding directional arrows between the components to show the flow of communication.

(3)



Range Of Computational Model

Sequential – instruction in an algorithm step by step

Parallel – processes are distributed between two or more processors. Each separate part of the algorithm that each processor processes can be combined together

In the parallel model, one task is processed by several processors while in the multi-agent model, separate tasks or algorithms are processed by different systems (agents) to perform a particular function.

SEQUENTIAL

In the sequential model, this involves following instructions in an algorithm step by step, in order, from start to finish.

PARALLEL

In the parallel model, computer processes are distributed between two or more processors in a computer with two or more processors installed.

Each separate part of the algorithm that each processor processes can be combined together. It requires an operating system capable of supporting two or more processors. It also requires software programs capable of distributing processes between them equally.

MULTI-AGENT

In the parallel model, one task is processed by several processors while in the multi-agent model, separate tasks or algorithms are processed by different systems (agents) to perform a particular function.

S2 What are the inputs, outputs and processing involved in uploading an image to a social networking profile?

Inputs: login details and other text entered via the keyboard, image captured by a camera/webcam.

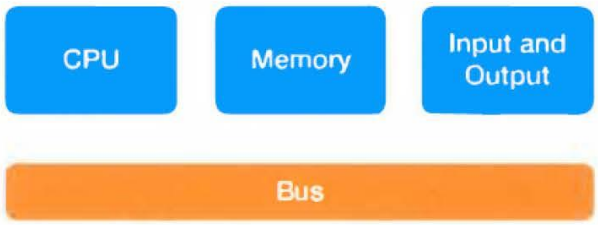
Outputs: image displayed on screen.

Processing: The computer would need to **authenticate** the user's **login details**, locate the appropriate web page, possibly **convert the image** to an appropriate file type and optimise it for display on the web, strip out any/all the metadata, **upload the image**.

Chapter 17 Hardware

1. Von Neumann Architecture / RAM / ROM/ CPU / Cache / VM
2. CPU Bus/ Fetch decode exe / CPU Performance Factors
3. Secondary Memory
4. Cloud Computing / Embedded System / IoT

van Neumann architecture *- computer system design in which the program is stored in memory with the data*



When the **CPU saves** data into **memory**, this is called **writing**

CPU uses the bus to tell the memory what data to save and where in the memory to save it.

The **reverse** process is called **reading**

Each memory location has a unique memory address

RAM	**ROM**
• Random Access Memory • Volatile Memory • When computer power off, data can be lost • Current instruction • gigabytes (GB)	• Read Only Memory • Non Volatile Memory • When computer power off, data can not be lost • Permanently stored – firmware • BIOS or UEFI

BIOS (Basic Input/Output System) or **UEFI** (Unified Extensible Firmware Interface).

UEFI has **replaced** BIOS as the standard firmware used in desktop computers

Cache Memory

In a CPU cache, **frequently used code or data is loaded** in chunks from the slower RAM into the cache.

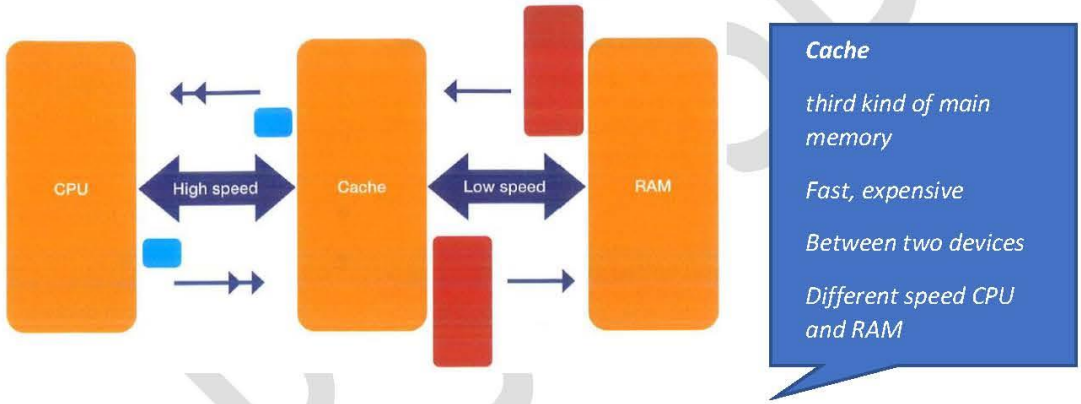
The CPU accesses the cache memory at its own, **faster speed.**

This means the CPU **isn't slowed down by having to wait for** data from the **RAM**.

- Between two devices - RAM and CPU
- Store frequently used data from RAM
- Speed up the processing

Cache memory is a small amount of fast, expensive memory that is used between two devices that communicate at different speeds, most often the CPU and RAM.

cache miss when the data requested for processing by a component or application is not found in the cache memory



ARM Vs Intel

ARM – RISC, Mobile

Intel – CISC, PC and laptop

Print spooling means **putting print jobs in a special area** called a buffer, in memory or on a disc.

VIRTUAL MEMORY

When a computer is running the OS and several applications at the same time, RAM becomes full

Instead of closing some program, memory manager of OS will use virtual memory – usually on hard disk drive

If there is no free memory, memory manager will 'swap out' some of the data stored in RAM

The least recently used stored data is swapped out

if needed again, it is swapped back

VM disadvantages

read/write speed of a hard drive is much slower than RAM

if the system has to rely too heavily on virtual memory, **significant drop in performance**.

Swap - **'disk thrashing**

more RAM a computer has, the **less virtual** memory

more RAM will **improve the performance** of a computer

CPU Bus - 3 types of bus

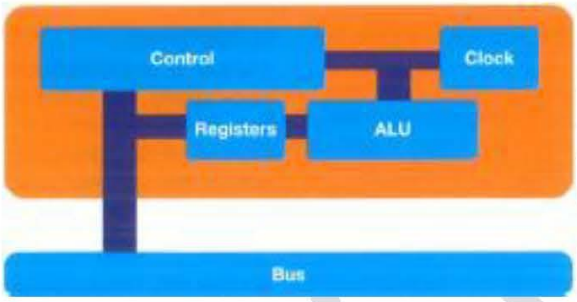
The bus that connects the CPU to other devices in the computer is split into **three parts**:

1. the address bus (unidirectional),
2. the data bus and
3. the control bus

The address bus **carries memory addresses between the CPU and memory to identify a memory location**;

the data bus **carries the value** to be read from or written to memory; and

the control bus **carries the signals** that, for example, determine whether to **read or write** the data, and when to do so.



The number of connections on a bus is called the **bus width**

greater bus width means **larger number values** can be communicated.

an **8-bit** address bus can **send** values from **00000000 (0) to 11111111 (255)**,

a **32-bit** computer can address up to **4 GB** of RAM.

FETCH-DECODE-EXECUTE CYCLE

Fetch The CPU control unit places the memory address of the next instruction on the address bus. It also sends a signal on the control bus requesting to read from memory. The memory receives the signal and looks up that memory location. The data in the memory is copied on to the data bus. The CPU copies this into a special register.

Decode The control unit analyses the contents of the register and sends signals to the other parts of the CPU telling them what to do (e.g. add numbers, store data back into the memory).

Execute The instruction is completed by the CPU.

COMMON REGISTERS

Accumulator: this holds the results of calculations performed by the ALU. All input and output from the CPU pass through the accumulator.

Program counter: this holds the memory address of the next instruction to be fetched.

Current instruction: this holds the instruction currently being executed.

FACTORS THAT AFFECT CPU PERFORMANCE

- **clock speed**
- **cache** size
- number of **cores**

Clock Speed

The rate at which instructions are processed by the CPU is controlled by the clock speed. ***The faster the clock speed, the faster the rate of processing***. **Disadvantages – overheat – Old Question**

Number of Cores

A multicore processor has more than one processor core (but only one CPU)

NUMBER OF CORES	COMMON NAME
1	Single-core
2	Dual-core
4	Quad-core
5	Penta-core
8	Octa(o)-core
10	Deca-core

Cache Size

With a larger cache there is a greater probability that the instruction or data item to be fetched is in the cache and so the RAM will not need to be accessed. This will speed up processing.

Most CPUs have independent **instruction and data** caches. The **data** caches have to be **read and written** to, but the **instruction caches** just have to be **read** by the CPU.

It does not have to wait for it to be fetched from the slower main memory. The **faster static RAM (SRAM)** is used for the cache.

The caches are **located on the processor** chip. The **fastest is the Level 1** cache and is **smaller** than the Level 2 and Level 3 caches.

The **Level 1 cache is checked first**, **followed by the Level 2** and then Level 3 caches. In a multicore processor, the cores have their own L1 and L2 caches while the Last Level cache is usually shared by all the cores.

With a **larger cache there is a greater probability** that the instruction or data item to be fetched is in the cache and so the RAM will not need to be accessed. This will **speed up processing**.

SECONDARY STORAGE

RAM is volatile. This means most computers need to be able to copy the contents of their RAM to another kind of storage that is not volatile - a type that doesn't lose its contents when there is no power. This more permanent storage is called **secondary storage**. Secondary storage is non-volatile and, compared to RAM, is slower to access, cheaper and has much higher storage capacity. Typical capacities are now in the terabytes, i.e. millions of megabytes.

Magnetic storage uses the fact that magnets have **north and south poles**. By making something **behave like a magnet**, the north and south poles can **represent the 1 s and 0s** of your data. This is used in hard disks and magnetic tape storage.

Optical storage is used by CDs and DVDs. **Shinier** or more reflective parts of the disk represent **the 1 s or 0s**.

Solid-state storage or 'flash' memory (such as **USB memory sticks** or **SD cards**) represents the **1 s and 0s** with little **pools of trapped electrons** on a microchip.

HDD - when data is read:

1. the arm moves across to be above the right track
2. the required sector comes around under the head
3. the surface behaving like a magnet causes a tiny current in the head
4. the disk controller translates this into 1s and 0s.

CD/DVD - when data is read:

1. The disk spins in the drive to ensure all data can be read.
2. The tracking mechanism moves the laser into the correct position over the disk.
3. The laser shines on to the disk and is reflected back on to a light sensor.
4. Signals from the sensor are translated into 1 s and 0s.

SDD - when data is read from the chip:

1. Control signals identify which bit is to be **read** out and apply a **small** voltage.
2. If the **electron pool is empty** the transistor turns on and a **1** is read out.
3. If the **electron pool is full** the transistor doesn't turn on and a **0** is read out.
4. The control signals are changed to read other bits.
5. When data is **written** to the chip, control signals identify which bit is to be written and apply a **higher** voltage.
6. This **pulls electrons** into the pools of those transistors, **recording the 1 s and 0s.**

Cloud Storage

Cloud storage **secondary storage**, often belonging to a **third party**, that is **accessed** via a **network**, usually the Internet, and so is **not in the same physical place** as the machine's **RAM/ROM**. Files stored 'in the cloud' can be **accessed from anywhere via an Internet** connection

Sometimes it can be useful to have your **secondary storage in a different place** from your computer.

Computers that share their secondary storage across a network are called **file servers**.

Sometimes storage is accessed via the Internet using Internet services such as Dropbox® or Google Drive. In this situation it is usually called **cloud storage**. This is an example of **virtualisation**. You seem to have access to a single, virtual hard drive when, in fact, your storage might be spread across several **servers around the world**, with software dealing with the details of where each part of each file is actually stored.

Advantages

- You can access the data from anywhere on many devices using a web browser.
- The data is securely backed up by the company providing the storage service.
- You don't need to transfer your data if you get a new computer.
- The cloud storage provider is responsible for the hardware your details stored on. They will need to ensure that the data is accessible and that they maintain the hardware and software needed to make the data available.
- The amount of storage available to an organisation can easily be changed as and when required - the cloud storage provider can normally make extra storage available in minutes. Doing this physically would mean ordering more hard drives and employing technicians to install them and set them up. This would take a lot longer than a few minutes.
- Having data stored off-site (not on the organisation's premises) means that it is protected from loss due to fire, theft of computers/servers, electrical failure, and so on. As the data is still available the business may be carried on from elsewhere. This could be the difference between staying in business or going bankrupt.
- Many cloud storage systems also manage the back-up of your data. They may take hourly or daily snapshots of your data and be able to restore from these in the case of hardware failure or accidental deletion of files.
- You are relying on a third-party storage provider to keep your organisation running. What happens if the cloud storage provider goes bankrupt, has a catastrophic failure or has its servers stolen? In any of these cases, it is likely you would be unable to access your data, at least temporarily, but possibly permanently.

Disadvantages

■Some people will be concerned about security: you might, for example, have read about high-profile cases of **private photographs** being accessed and released on the Internet.

■Other people might be uncomfortable with not being able to have their data 'in their hand'.

■Data stored anywhere accessible via the Internet carries the risk of other people gaining access to it. This might happen through a deliberate attack by hackers, or as a result of an accidental error in the way it is set up that results in the data becoming available to the public. Also, the Internet isn't a particularly secure network. Data may be accessed as it is being sent to/from the cloud storage provider. The loss of data, particularly of a confidential nature, could be devastating for an organisation.

■The cloud storage provider might have its servers located in a different country (or countries) to your organisation. Some types of data can only legally be stored in countries that have similar data protection laws to our own.

■Access to cloud storage is dependent on having a reliable, high-speed Internet connection available. How will a business operate if the Internet becomes unavailable for some reason?

Embedded System

one that is designed to do a specific job such as a washing machine

all around us in every part of daily life such as car, games console, traffic lights

cheap, low-power devices using simple technology and limited memory

rarely updated

stored permanently in firmware/ use sensors

Embedded systems in their daily lives include: TV, remote control, fridge, microwave, washing machine, oven, electronic toys, car, sports band, traffic lights, central heating, burglar alarm.

Internet of things (IoT)

The **Internet** and **embedded** systems have come together (converged).

Many devices can now communicate with each other and be **accessed and controlled remotely via** the Internet.

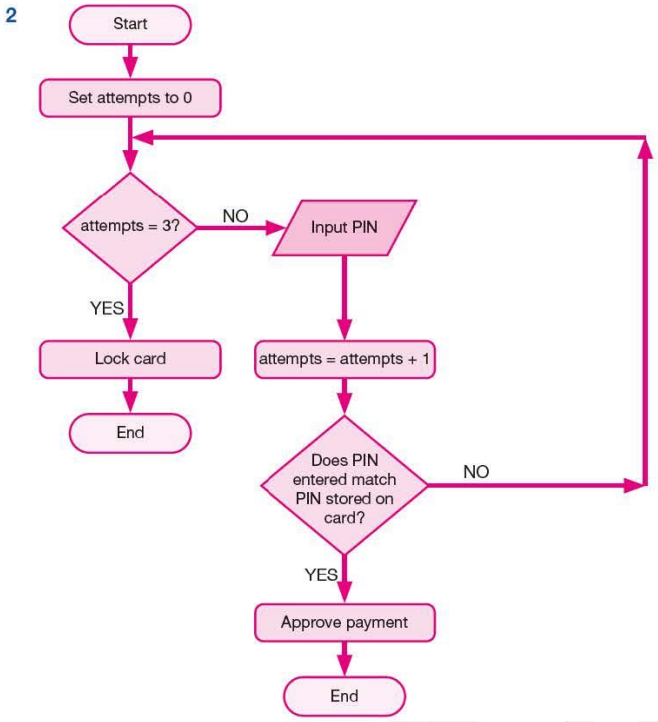
This has become known as the 'Internet of things'.

Example

- your **printer** in school might **automatically order new toner** when it is getting low
- your **electricity meter** might send readings straight to the supplier via the Internet
- you might be able to **control your central heating** from your smartphone.
- Imagine a future where your **fridge knows** when you're **running out of milk** and sends you a text telling you to pick up some more on the way home.

Draw a flowchart showing the processing required by **a chip-and- PIN credit card reader**. This is used when you pay for something in a shop. It collects data about the purchases from the till, gets your card details, checks your PIN, sends encrypted data to the bank and waits for a reply. It might print a receipt if the purchase is allowed and informs the till of the result.

2



What are the advantages and disadvantages of different kinds of secondary storage?

Data is written to and read from a magnetic hard disc more quickly than to/from an optical disc, so backing up and restoring takes less time.

Hard discs are permanently located within a drive so are less portable than optical discs

External hard drives are very light and compact so are reasonably portable. That said, an external hard drive still has moving parts, which might get damaged

Optical media tends to be more durable than magnetic media.

Magnetic hard drives have now been replaced by solid-state drives in many laptops.

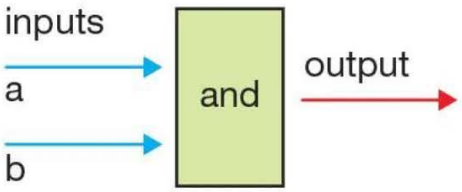
Solid-state drives are faster, lighter, quieter and more resilient than magnetic hard drives.

More expensive and have a smaller capacity

Chapter 18

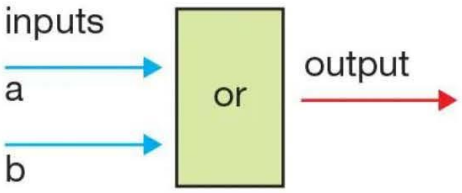
Logic circuits

AND



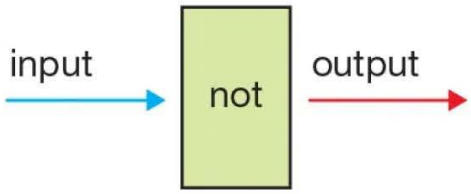
inputs		output
a	b	
0	0	0
0	1	0
1	0	0
1	1	1

OR



inputs		output
a	b	
0	0	0
0	1	1
1	0	1
1	1	1

NOT



input	output
0	1
1	0

Table 4.1 Truth table for the AND operator

IF health<=0 AND live=0 THEN

SET Game Over TO True

END IF

Input ➔ health<=0 / lives=0 ➔ 2 inputs ➔ 2^2= 4 possible outcomes

Health<=0	Lives=0	Game over
0	0	0
0	1	0
1	0	0
1	1	1

HEALTH <= 0	LIVES = 0	GAME OVER?
No	No	No
No	Yes	No
Yes	No	No
Yes	Yes	Yes

Table 4.2 Truth table for the OR operator

IF health <= 0 OR score > 1000000 THEN

SET game_over TO true

END IF

Health<=0	Score>1000000	Game over
0	0	0
0	1	1
1	0	1
1	1	1

Table 4.4 Truth table for showing all possible combination

IF (NOT god_mode AND health <= 0) OR score > 1000000 THEN

SET game_over TO true

END IF

Input ➔ got_mode / health<=0 / score>1000000 ➔ 3 inputs ➔ 2^3= 8 possible outcomes

God_mode	Health<=0	Score>1000000	NOT god_mode	Not god_mode AND health<0	Game_over
0	0	0	1	0	0
0	0	1	1	0	1
0	1	0	1	1	1
0	1	1	1	1	1
1	0	0	0	0	0
1	0	1	0	0	1
1	1	0	0	0	0
1	1	1	0	0	1

ACTIVITY 17

LOGIC AND CONDITIONS

1 In words, write the condition for the game to be over in the final example in the table. (You will see why Boolean logic is clearer and easier to use than English when the situation gets complicated.)

2 Which part of the CPU works out the logic for conditions?

3 Create a computer program that asks for your year group, grade and target and then prints out whether you are invited to revision class or not. Start by assuming revision classes are only for Year 11 students whose grade is below target, then you can invent your own criteria for invitations to revision class.

4 Look at the statement below and write the truth table. This time there are three parts to the condition. Remember: as in maths, brackets are worked out first.

```
IF year = 11 AND (grade < target OR target > 7) THEN
    SET revision_class TO true
END IF
```

1 The game is over either when the player's score exceeds 1 million or if the player runs out of 'health' and isn't operating in 'god mode'.

2 ALU

3 Here is the 'starter' program written in Python:

```
yearGroup=int(input("Enter year group"))
grade=int(input("Enter grade (9-1)"))
target=int(input("Enter target grade (9-1)"))
if yearGroup==11 and grade<target:
   print("You should attend the revision class")
else:
   print("No need to attend the revision class.")
```

Grade<target	Target>7	Year=11	(Grade<target OR Target>7)	Year =11 AND (Grade<target OR Target>7)
0	0	0	0	0
0	0	1	0	0
0	1	0	1	0
0	1	1	1	1
1	0	0	1	0
1	0	1	1	1
1	1	0	1	0
1	1	1	1	1

Worked Example

*If plants have **yellow** leaves mainly near the soil, then they need a **nitrogen** fertiliser.*

*If plants have **brown** leaves that are small, then they need a **phosphorous** fertiliser.*

```
SET advice TO 'nothing need'
IF colour='yellow' THEN
        SET advice TO 'nitogen'
ELSE
        IF colour='brown' THEN
                SET advice TO 'phosphorous'
        END IF
END IF
```

*Mohammad points out that if leaves go **brown**, but are **normal** sized, then the customer needs **potassium** fertiliser. Now, the size of the leaf is also important.*

```
SET advice TO 'nothing'
IF colour='yellow' THEN
        SET advice TO 'nitogen'
ELSE
        IF colour='brown' AND size='small' THEN
                SET advice TO 'phosphorous'
        ELSE
                IF colour='brown' AND size='normal' THEN
                        SET advice TO 'potassium'
                END IF
        END IF
END IF
```

ACTIVITY 18

ABSTRACTION

1 Add the following rules to Mohammad's website logic in the example.

- If plants have cracked leaves or leaves that are the wrong shape, then they need a calcium fertiliser.
- If plants have leaves that are yellow mainly on the tips then they need a magnesium fertiliser.

2 Why is Mohammad's advice an example of abstraction?

```
SET advice TO 'nothing'
IF colour='yellow' THEN
        IF leaftips=TRUE THEN
                SET advice TO 'magnesium'
        ELSE
                SET advice TO 'nitogen'
        END IF
ELSE
        IF colour='brown' AND size='small' THEN
                SET advice TO 'phosphorous'
        ELSE
                IF colour='brown' AND size='normal' THEN
                        SET advice TO 'potassium'
                ELSE
                        IF leaves='cracked' OR leaves='misshapen' THEN
                                SET advice TO 'calcium'
                        END IF
                END IF
        END IF
END IF
SEND advice TO DISPLAY
```

Worked Example

Nadia has built her own programmable alarm clock. She wants to program the alarm to wake her up at the correct time depending on the day. She gets up:

- at 7.30 a.m. if it is a school day
- at 9.00 a.m. if it is the weekend or a school holiday, except on Saturdays in term time when she gets up at 8.00 a.m. to play hockey.

```
SET alarm TO '7:30 am'
IF term_time = false OR (day = 'Saturday' OR day = 'Sunday')
THEN
    SET alarm TO '9:00 am'
ELSE
    IF term_time = true AND day = 'Saturday' THEN
        SET alarm TO '8:00 am'
    END IF
END IF
```

```
SET alarm TO '7:30 am'
IF term_time=true OR day='Saturday' THEN
        SET alarm TO '8:00 am'
ELSE
        IF term_time=false OR (day='Saturday'  OR day='Sunday') THEN
                SET alarm TO '9:00 am'
        END IF
END IF
```

Chapter 19

Computer systems consist of both hardware and software.

hardware consists of items that we can see and touch such as keyboards, disk drives and monitors.

Software is the set of programs that run on a computer system. Two types of software are Application Software and System Software

Application software software that performs a task that would otherwise be done by hand, perhaps with pen and paper

System software operating system and utility software

operating system software designed for particular hardware and which manages other programs' access to the hardware

utility software software that does a **useful job** for the user that is **not essential to the operating** system

OQ – OS – store data in HDD – step by step procedures

OS major tasks

Managing files and the directory structure, and **input/output**, is one of the important jobs of the operating system.

An operating system also **shares access to the hardware** among the different programs that are running

Two **important pieces of hardware** that have to be shared are the **CPU and RAM**.

Scheduling

When you have **several applications open**, they can't really run at the same time. The operating system keeps up this illusion by allowing each program to use the CPU for a **short time before switching** to the next program. This is called **scheduling**.

How much the hardware is used by each process is displayed, including:

- the **proportion of the CPU time spent** on this process
- the **amount of memory used** by this process
- the **amount of available memory**

Multitasking

As well as allowing **multitasking**, some users operating systems allow different to use the hardware.

In some cases, only one user can use the hardware at once (single user).

In others, several users can be sharing the hardware (multi-user).

Utility software

does a useful job / not essential to the operating system

three areas

1. basic tools
    a. simple text editor
    b. calculator
    c. command prompt
2. file management
    a. recover file – back_up copy
    b. format factory
    c. disk defragment
3. security.
    a. 1. recover file – back_up copy
    b. 2. format factory
    c. 3. disk defragment

Simulation and Modeling

Prediction outcomes - using computer models to predict what might happen in the future

two big problems

The model or simulation includes **assumptions**. It is not reality, so the **answers might not be right**

Import/ input the incorrect / inaccurate data into the model, the output might not be right.

Why might some users prefer a **command-line interface** to a graphical one?

programmar

Why is antivirus software utility software?

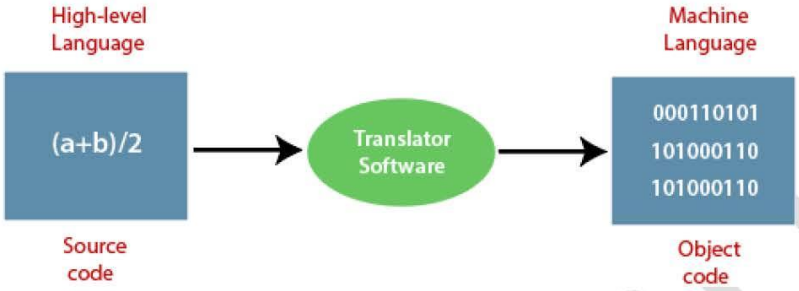
Anti-virus software is not essential to the operation of the operating system and isn't an application in its own right.

Chapter 20

algorithm can be written in any language that a human can understand

computers can only understand instructions written as a series of 1s and 0s – machine code

code programs can be translated into machine code



High Level Programming Languages - python

Low Level Programming Languages – assembly

Translator

Translators are also programs.

Their input is the text - source code. Their output is machine code.

simplest translator is called an assembler

converts assembly language to machine code

Assembly language is called a low-level programming language

Three Types of Translators

Assembler	**Compiler**	**Interpreter**
low-level programming	High Level Programming Languages	High Level Programming Languages
converts assembly language to machine code	translator that converts high-level language source code into object code	translator that converts high-level language source code into object code
Assembly language is a bit easier to work with than machine code (mnemonic – ADD)	translator that translates the **whole** program	interpreter translates your source code **one line at a tim**e
challenging for three reasons:	.exe file extension Easier to protect code	No object code file Run Slower

a. A very limited range of instructions is available b. have to decide how to represent your data c. Debugging is very difficult		

INTERPRETER	COMPILER
Every computer that will run your program needs the interpreter software installed.	The output from a compiler will run on its own on any similar computer.
Interpreters find errors when they happen and can often tell you what has gone wrong.	A compiler cannot produce any object code unless the whole program is correct – they tend to report a lot of errors initially, making it harder to debug your program.
Programs tend to run slower using an interpreter because the interpreter has to translate the source code while the program is running.	It can be easier to protect your code from being altered or copied if it has been compiled because you only give people the object code (machine code), which is hard to understand.