ALevel CS C16 Data representation

© 2021,有鱼英语
https://collapsar.yuketang.net/
https://coding.yuketang.net/

Data types

Build-in data types: programming language defines **the range of possible values** and **the operations available for manipulating**

User-defined data types: the programmer has included the definition in the program.

1. create a new datatype
2. allow data types not available in a programming language

```
TYPE <TypeIdentifier>
   DECLARE <field identifier> : <data type>
ENDTYPE
DECLARE <variable identifier> : <record type>
```

Non-composite data types: does **not involve a reference** to another data type.

Enumerated data type: an example of a **user-defined non-composite** data type.

```
TYPE
TDirections = (North, East, South, West)
DECLARE Direction1 : TDirections
Direction1 ← North
```

Composite user-defined data types: **reference to at least one other type**.

record data type and class

```
DECLARE
TYPE <type_name> IS RECORD
(
   <column1> <datatype>,
   ...
   ...
)
```

Point data type: **reference** to a memory location,

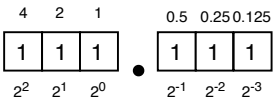
```
TYPE TIntegerPointer ← ^Integer
DECLARE MyIntegerPointer : TIntegerPointer
DECLARE Number1, Number2 : INTEGER
Number1 ← 100
MyIntegerPointer ← @Number1
Number2 ← MyIntegerPointer^ * 2
```

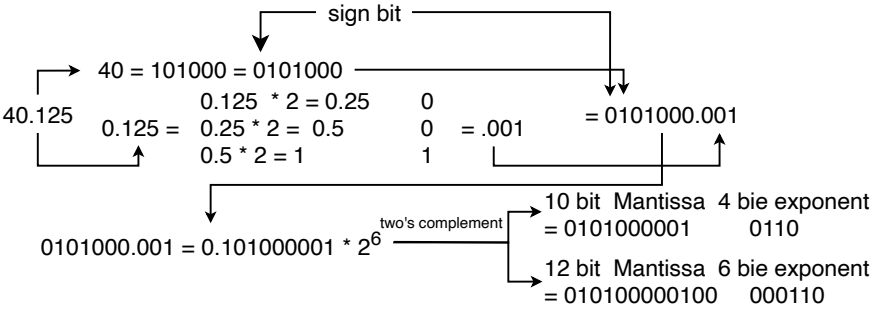
Set data type:

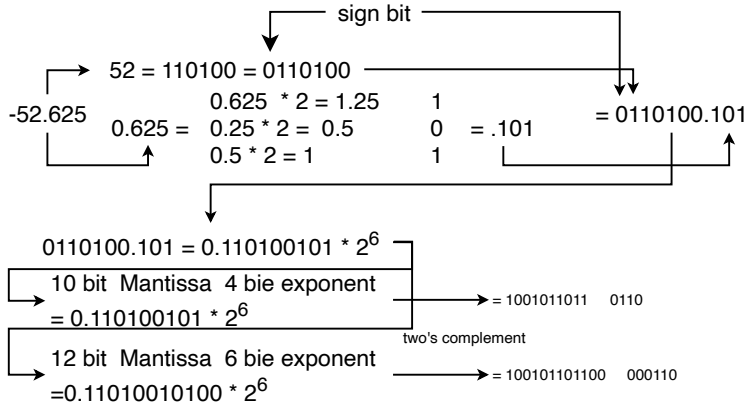
a collection of data items that lacks any structure; contains **no duplicates** and has a number of defined operations that can be performed on it

Convert real number to binary number

Size of mantissa decreased - precision is reduced
Size of exponent increased - range is increased







Precision

real number in program language precision error :
1. 0.1+0.1+0.1 != 0.3
2. round(2.5) = 2

reason:

1. real number cannot be represented exactly in binary
2. the manipulate will increase the precision.
3. difference after the calculation is significant enough to be seen

File Organisation

Binary file: a file designed for storing data to be used by a computer program

Record: a **collection of fields** containing data values

text file: contains data stored according to a character code of the type

binary file: stores data in its internal representation

Serial files: contains records that have **not been organised in any defined order**. A database that stores values as a series of items, one after the other

Sequential files: records that are **ordered**

Direct-access files: only that the access can be to any record in the file **without sequential reading** of the file

1. use a sequential search to look for a vacant address following the calculated one
2. keep a number of overflow addresses at the end of the file
3. have a linked list accessible from each address.

Convert binary number to real number

Mantissa 00101000 0.0101000 0010.1000 2.5
Exponent 00000011 3

-1 1's c shift 1
Mantissa 1001 1.001 ~ 1.000 ~ 0.111 ~ 01.11 ~ -1.75
Exponent 0001 1

Mantissa 10110000 -1 1's complement shift 2 places
1.0110000 ~ 1.0101111 ~ 0.1010000 ~ 0.001010000 ~ -5/32

Exponent 1110 1101 ~ 0010 ~ -2

Normalization

For a positive number, the bits in the mantissa are shifted left until the most significant bits are 0 followed by 1. For each shift left the value of the exponent is reduced by 1. The same process of shifting is used for a negative number until the most significant bits are 1 followed by 0. In this case, no attention is paid to the fact that bits are falling off the most significant end of the mantissa.

Mantissa 0000000111 0.000000111 ~ 0.111 (6)
Exponent 100111 100111 ~ 100110 ~ 011001 ~ -25 - 6
= -31~ 0111101~0111110~100001

Mantissa 110100 1.10100 ~ 1.0100 (1)
Exponent 0010 0010 ~ 2 - 1 = 1 ~ 0001

problem not normalization:

1. **precision lost**
2. Redundant leading zeros in the mantissa
3. **Bits lost off right hand end / least significant end**
4. **Multiple representations of a single number**

Overflow

Overflow - Overflow occurs when calculations produce results exceeding the capacity of the result.

Underflow - A calculation resulting in a number so small that the negative number used for the exponent is beyond the number of bits used for exponents is called underflow

Description	BinaryCode	Denary equivalent
Largest positive value	0 111 0111	$0.875 * 2^7 = 112$
Smallest positive value	0 001 1000	$0.125 * 2^{-8} = 1/2048$
Smallest magnitude negative value	1 111 1000	$-0.125 * 2^{-8} = -1/2048$
Largest magnitude negative value	1 000 0111	$-1 * 2^7 = -128$

Problem

1. the conversion of a real value in denary to a binary representation almost guarantees a degree of approximation.
2. There is also a restriction of the number of bits used to store the mantissa.

ALevel CS C18 Hardware and virtual machine

© 2021,有鱼英语
https://collapsar.yuketang.net/
https://coding.yuketang.net/

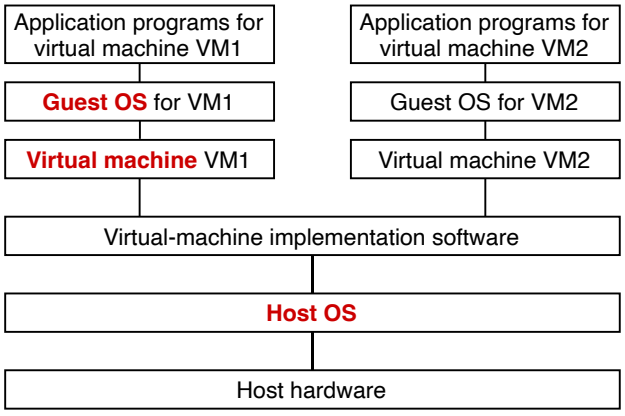
Control Unit

Control Unit: to ensure that each machine instruction is handled correctly
Two methods control unit can be designed:
1. **logic circuit**: hardware solution. The machine-code instructions are handled directly by hardware.
2. **microprogramming**: contains a ROM component that stores the microinstructions or microcode for microprogramming, often referred to as firmware.

CISC and RISC processor

Complex Instruction Set Computer (CISC): a single instruction can be more complex and involve more loading of data from memory
Reduced Instruction Set Computer (RISC): a single instruction is simpler, requiring minimal loading of data from memory
CISC and RISC point:
1. For RISC the term 'reduced' affects more than just the number of instructions.
2. A reduction in the number of instructions is not the major driving force for the use of RISC.
3. The reduction in the complexity of the instructions is a key feature of RISC.
4. The typical CISC architecture contains many specialised instructions.
5. The specialised instructions are designed to match the requirement of a high-level programming language.
6. The specialised instructions require multiple memory accesses which are very slow compared with register accesses.
7. The simplicity of the instructions for a RISC processor allows data to be stored in registers and manipulated in them with no resource to memory access other than that necessary for initial loading and possible final storing.
8. The simplicity of RISC instructions makes it easier to use hard-wiring inside the control unit.
9. The complexity of many of the CISC instructions makes hard-wiring much more difficult so microprogramming is the norm.

RISC	CISC
Fewer instructions	More instructions
Simpler instructions	More complex instructions
Small number of instruction formats	Many instruction formats
Single-cycle instructions whenever possible	Multi-cycle instructions
Fixed-length instructions	Variable-length instructions
Only load and store instructions to address memory	Many types of instructions to address memory
Fewer addressing modes	More addressing modes
Multiple register sets	Fewer registers
Hard-wired control unit	Microprogrammed control unit
Pipelining easier	Pipelining more difficult



Pipelining

Pipelining: instruction-level parallelism. a technique used to **improve the execution throughput of a CPU** by using the processor resources in a more efficient manner

Colck cycle							
	1	2	3	4	5	6	7
Instruction fetch (IF)	1.1	2.1	3.1	4.1	5.1	6.1	7.1
Instruction decode(ID)		1.2	2.2	3.2	4.2	5.2	6.2
Operand fetch(OF)			1.3	2.3	3.3	4.3	4.3
Instruction execute(IE)				1.4	2.4	3.4	4.4
Result write back(WB)					1.5	2.5	3.5

Pipelining for five-stage instruction handling:
1. For pipelining to be implemented, the construction of the processor must have **five independent units**, with each handling one of the five stages identified.
2. Once under way, the pipeline is handling five stages of five individual instructions.
3. One issue with a pipelined processor is **interrupt handling**.
   1. erase the pipeline contents for the latest four instructions to have entered.
   2. construct the individual units in the processor with individual program counter registers.

Two issue will cause the pipeline to stall:
1. dealing with a **data dependency between instructions**
2. **branch instructions**

CISC and RISC processor

Parallel processing: means that the architecture has more than one processor. Different processors are responsible for different parts of tasks.
SISD: Single Instruction Stream Single Data stream; a single processor accessing one memory
SIMD: Single Instruction Stream Multiple Data stream; processing of parallel data input requiring one control unit instructing multiple processing units
MISD: Multiple Instruction Stream Single Data stream; does not exist in a single architecture
MIMD: Multiple Instruction Stream Multiple Data stream; multiple processors asynchronously processing parallel data input
Massively parallel computer systems:
1. a network infrastructure to support multiple computer units.
2. The programs running on the different computers can communicate by passing messages using the network
3. cluster computing using PCs
4. an extremely large number of individual processors working in parallel.

These are the systems used by large organisations for computations involving highly complex mathematical processing
Issue:
1. Communication between the different processors is the issue
2. Each processor needs a link to every other processor
3. Many processors require many of these links
4. Challenging topology

Virtual Machine

System virtual machine: the emulation of computer system hardware using software.
Virtual machine advantage:
1. more than one different operating system can be made available on one computer system
2. an organisation has legacy systems and wishes to continue to use the old software but does not wish to keep the old hardware.
3. the same operating system can be made available many times by companies with large mainframe computers that offer server consolidation facilities.

Virtual machine disadvantage:
1, the time and effort required for implementation
2. low performance

ALevel CS C17 Communication and Internet (1)

© 2021,有鱼英语
https://collapsar.yuketang.net/
https://coding.yuketang.net/

Transmissions models(Circuit switching)

Circuit switching: the method used in the **traditional telephone system**
Public Switch Telephone Networks(PTSNs)**:** largely converted to digital technology. the same method can be provided for data transfer that was traditionally used for voice communication
Circuit switching step:
1. The sender provides the identity of the intended receiver.
2. The system checks whether or not the receiver is ready to accept data.
3. If the receiver is available, a sequence of links is established across the network.
4. The data is transferred.
5. The links are removed.

Circuit switching benefits:
1. **no delay**: delay is minimized and no switching
2. **all bandwidth**: provide with consistent bandwidth, channels, and an ongoing data rate
3. **data in order**: data packets are delivered in their correct order
4. **not sharing channel**

Circuit switching drawback:
1. Dedicating one channel to a single service leaves it unavailable to other services.
2. It is expensive to provision an entire channel to one service and one individual routing path.

Usage: telephone, video conference, live streaming.

Comparison between circuit and packet switching	
Circuit switching	packet switching
Dedicated path	No dedicated path
Path is established for entire conversation	Route is established for each packet
Call setup delay	Packet transmission delay
Overload may block call setup	Overload increases packet delay
Fixed bandwidth	Dynamic bandwidth
No overhead bits after call setup	Overhead bits in each packet

Protocols

Protocol: a set of rules for data transmission which are agreed by sender and receiver
Protocol suite:
1. Each layer can only accept input from the next higher layer or the next lower layer.
2. There is a defined interface between adjacent layers which constitutes the only interaction allowed between layers.
3. A layer is serviced by the actions of lower layers.
4. With the possible exception of the lowest layer the functioning of a layer is created by installed soft ware.
5. A layer may comprise sub-layers.
6. Any user interaction will take place using protocols associated with the highest level layer in the stack.
7.Any direct access to hardware is confined to the lowest layer in the stack.

Why necessary:
1. All data is using the **same rules**
2. All data is using the **same formats**
3. allow communication between devices operating on **different platforms**
4. The communication is independent of the software used
5. The communication is independent of the hardware used

Transmissions models (Packet switching)

Packet Switching:
1. Allows data transmission without a circuit being establish.
2. Data can be sent is divided into packets.
3. Data can travel along different routes.
4. Packets are reassembled in the correct order at the receiver's end
5. Wait until last packet is received, then put the data back together

A packet consists of a **header** which contains instructions for delivery plus the **data body**.

there are two ways that the network can provide a service: **connectionless service** or **connection-oriented service**.
Connectionless service: a packet is dispatched with no knowledge of whether or not the receiver is ready to accept the packet, and has no way of finding out if the transmission has succeeded.
Connection-oriented service: the first packet sent includes a request for an acknowledgement.If the acknowledgement is received, the sender transmits further packets. If no acknowledgement is received, the sender tries again with the first packet. Packet switching are mainly used for data and voice applications requiring non-real time scenarios

Packet switching benefits:
1. **Accuracy**: Ensures accurate delivery of the message
2. **Completeness**: Missing packets can be easily detected and a re-send request sent so the message arrives complete
3. **Resilience**: if a network changes the router can detect this and send the data another way to ensure it arrives
4. **Path also available to other users** // Doesn't use whole bandwidth // allows simultaneous use of channel by multiple users

Packet switching drawback:
1. Time delays to correct errors // Network problems may introduce errors in packets
2. Requires complex protocols for delivery
3. Unsuitable for real time transmission applications

Why packet switching is used in internet?
1. Packet switching makes best use of the available (channel) capacity
2. by using alternative routes
3. which is more secure / robust
4. as packets to / from different sources and destinations can share the same route

Packet switching feature:
1. In packet switching, **station breaks long message into packets**. Packets are sent one at a time to the network. Packets are handled in two ways, viz. datagram and virtual circuit.
2. **In datagram**, each packet is treated independently. **Packets can take up any practical route**. Packets may arrive **out of order** and **may go missing**.
3. **In virtual circuit**, preplanned route is established before any packets are transmitted. The **handshake** is established using call request and call accept messages.In this type, **routing decisions for each packet are not needed**.
4. For a packet-switched network, data is transferred by dividing the data into individual packets and passing it through the circuits to the other host. In packet -switched networks, the route is not exclusively determined when the packets hit the wire. Using routing algorithms, **each packet may actually take a different route** through the network to arrive at the destination host.

TCP/IP protocol suite

TCP/IP protocol suite: TCP/IP is the protocol suite underpinning the Internet usage.
1. **Application layer**: handless access to service, manage data exchange, defines protocol used
HTTP,SMPT,DNS,FTP,POP3,IMAP
2. **Transport layer**: handles the forwarding of packets
TCP,UDP,SCTP
3. **Network(internet) layer**: handles transmission data, routing, IP address.
IP,IGMP,ICMP,ARP
4. **Link layer(physical layer)**: handles how data is physically send

ALevel CS C17 Communication and Internet (2)

© 2021,有鱼英语
https://collapsar.yuketang.net/
https://coding.yuketang.net/

TCP: The TCP protocol operating in the transport layer now has to take responsibility for ensuring the safe delivery of the 'message' to the receiver. Each packet consists of a header plus the user data.

1. **ensure safe delivery**
2. **ensure that any response is directed back** to the application protocol.
3. **connection-oriented**

Tcp function:

1. allows applications to exchange data
2. establishes and maintains a connection
3. until exchange of data is complete
4. determines how to break application data into packets
5. adds sequence / packet number to (TCP) header
6. sends packets to and accepts packets from the network / Internet layer
7. manages flow control // manages congestion avoidance
8. acknowledges all packets that arrive
9. detects when a packet has not arrived at destination
10. handles retransmission of dropped packets
11. reassembles packets into the correct order

TCP/IP data packet have following information:

1. **source** - which computer the message came from
2. **destination** - where the message should go packet
3. **sequence** - the order the message data should be re-assembled
4. **data** - the data of the message
5. **error check** - the check to see that the message has been sent correctly

Route:

1. the frame sent by the data-link layer will arrive at a router during transmission (more likely at several routers).
2. the function of the router software to choose the next target host in the transmission.
3. The routing table for every router has details of any current problems with any of the options for the next transmission step.

The major **distinction** between a switch and a router as a node in a network is that when a frame arrives at a switch, it is transmitted on without any routing decision. A switch operates in the data-link layer but has no access to the network layer.

routing table: network id, routing metric, next hop, interface

IP(Internet Protocol): to ensure correct routing over the Internet. IP protocol takes the packet received from the transport layer and adds a further header. The header contains the IP addresses of both the sender and the receiver.

IP data packet content: IP version number, internet header, total length, protocol, source ip, destination ip address, checksum

IP functions:

1. routes the packets around the network
2. adds to the IP header a source/destination address for each packet
3. encapsulates data into datagram
4. passes datagram to the network access layer (for transmission on the LAN)// passes datagram to the transport layer (on arrival at destination)
5. Defines the addressing method e.g. subnetting, NAT

datagram: The IP packet, which is usually called a 'datagram', is sent to the data-link layer and therefore to a different protocol suite.

IP functions as a **connectionless service**

HTTP(HyperText Transfer Protocol): transfer web pages from servers to the client.

1. transaction-oriented, client-server protocol.
2. define the format of message
3. HTTPS is Hyper Text Transfer Protocol Secure when data is encrypted and send with security.

FTP(File Transfer Protocol): the application-layer protocol that can handle any file transfer between two end-system.

Email protocol

SMTP(simple Mail Transfer Protocol): push protocol, sending a mail

1. largely replaced by the use of web-based mail

POP3(Post Office Protocol version3): pull protocol

1. for emails to be downloaded onto the client computer
2. more secure to cyber-attack because emails are locally stored
3. only accessible from one client system.

IMAP(Internet Message Access Protocol):

1. not download the email to client computer
2. the server can be accessed from any client

P2P (Peer-to-peer) file sharing: no structure and no controlling mechanism. Peers act as both clients and servers and each peer is just one end-system. When a peer acts as a server it is called a 'seed'.

how the bitTorrent protocol allows file to be share:

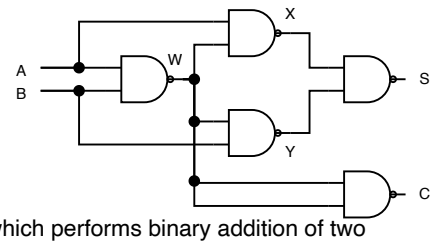
1. BitTorrent client software made available
2. One computer must keep a complete copy of the torrent/file to be shared
3. Torrent/file is split into small pieces
4. A computer joins (a swarm) by using the BitTorrent software to load a torrent descriptor file
5. The computer can now download a piece of the file
6. Once a computer has a piece it can become a seed and upload (to other members of the swarm)
7. Pieces of the torrent are both downloaded and uploaded (by each member of the of the swarm)
8. A server called a tracker keeps records of all the computers in the swarm
9. The tracker shares their IP addresses allowing them to connect to each other

ALevel CS C19 Logic Circuits and Boolean algebra

© 2021,有鱼英语
https://collapsar.yuketang.net/
https://coding.yuketang.net/

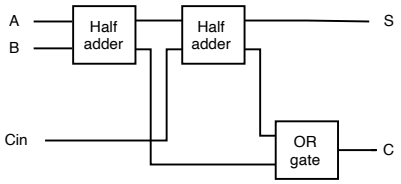
The half adder: a circuit which performs binary addition of two individual bits
sum bit(S) and **carry bit (C)**

Input		Output	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



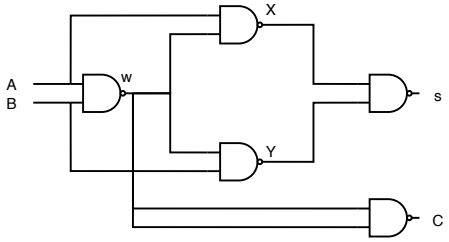
The full adder: a circuit which performs binary addition of two individual bits and an input carry bit. **carry in bit (Cin)**

A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	0
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



Boolean algebra: a circuit in which the output is dependent only on the input values

Identity/Law	AND form	IR form
Identity	$1.A = A$	$0 + A = A$
Null	$0.A = 0$	$1 + A = 1$
Idempotent	$A.A = A$	$A + A = A$
inverse	$A.\overline{A} = 0$	$A + \overline{A} = 1$
Commutative	$A.B = B.A$	$A + B = B + A$
Associative	$(A.B).C = A.(B.C)$	$(A+B)+C = A+(B+C)$
Distributive	$A+B.C = (A+B).(A+C)$	$A.(B+C) = A.B+A.C$
Absorption	$A.(A+B) = A$	$A + A.B = A$
De Morgan's	$\overline{A.B} = \overline{A} + \overline{B}$	$\overline{A+B} = \overline{A}.\overline{B}$
Double Component	$\overline{\overline{A}} = A$	$\overline{\overline{A}} = A$



Step 1 NAND Truth Table

NAND Truth Table		
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

Step 2 Truth Table -> Logic Expression

$$W = \overline{A}.\overline{B} + \overline{A}.B + A.\overline{B}$$

Step 3 Logic Expression X and Y

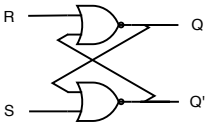
$$X = \overline{A.W}$$
$$X = \overline{A.(\overline{A}.\overline{B} + \overline{A}.B + A.\overline{B})}$$
$$X = \overline{0 + 0 + A.A.\overline{B}}$$
$$X = \overline{A.\overline{B}}$$
$$X = A + \overline{B}$$
$$Y = \overline{A} + B$$

Step 4 Logic Expression S

$$S = \overline{X.Y}$$
$$S = \overline{X} + \overline{Y}$$
$$S = \overline{(A + \overline{B})} + \overline{(\overline{A} + B)}$$
$$S = \overline{A}.B + A.\overline{B}$$

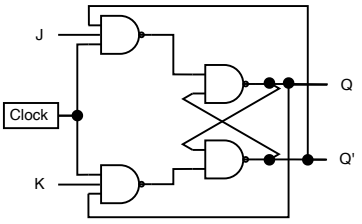
combinational circuits: a circuit in which the output is dependent only on the input values
Sequential circuit: a circuit in which the output depends on the input values and the previous output
SR flip-flop:
S - set R - reset
set state(self-consistent): Q = 1, Q' = 0
unset state: Q = 0, Q' = 1
These properties explain why the SR flip-flop can be used as a storage device for 1 bit and therefore could be used as a component in RAM because a value is stored but can be altered.

Input signals		Initial state		Final state	
S	R	Q	Q'	Q	Q'
0	0	1	0	1	0
1	0	1	0	1	0
0	1	1	0	0	1
0	0	0	1	0	1
1	0	0	1	1	0
0	1	0	1	0	1



JK flip-flop:
difference: if both J and K are input as a 1 then the values for Q and Q' are toggled (they switch value).
This makes the JK flip-flop a more reliable device because there is no combination of input states that leave uncertainty as to which values are stored.

Input signals		Initial state	Final state
J	K	Clock	Q
0	0	↑	Q unchanged
1	0	↑	1
0	1	↑	0
1	1	↑	Q toggles



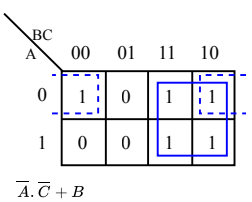
difference SR & JR:
1. SR flip-flop has an invalid combination of S and R. SR allow both Q and Q' to have the same value. SR inputs may arrive at different times.
2.JK flip-flop all four combination of values for J and K are valid.JK does not allow Q and Q' to have the same value.JK incorporates a clock pulse for synchronisation

The role of flip-flop in a computer:
1. A flip-flop can store either a 0 or a 1
2. Computers use bits to store data
3. Flip-flops can therefore be used to store bits (of data)
4. Memory can be created from flip-flops

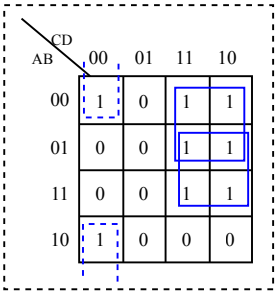
Karnaugh maps(K-maps): a method of creating a Boolean algebra expression from a truth table.
A K-map can make the process much easier than if you use sum-of-products to create minterms. If applied correctly a K-map produces the simplest possible form for the Boolean algebra expression.

A	B	C	X
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

$$\overline{A}.\overline{B}.\overline{C} + \overline{A}.B.\overline{C} + \overline{A}.B.C + A.B.C$$



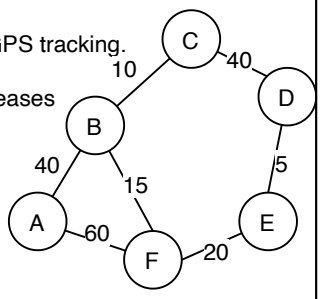
$\overline{A}.\overline{C} + B$



ALevel CS C21 Artificial Intelligence(AI)

© 2021,有鱼英语
https://collapsar.yuketang.net/
https://coding.yuketang.net/

Graphic: a collection of nodes or vertices between which there can be edges. Each **node** has a name. An **edge** can have an associated label which is a numerical value.
Dijkstra's algorithm:finds the shortest path to each of the other nodes starting from one of the nodes.
Structure English:
Identify the source node (S) where the path starts.
Create an empty set called the ShortestPath set.
Create another set called RemainingNodes and put all of the nodes into this including the source node (S).
Create a record that stores:
node names
calculated values for the distance to the node from the source node
the sequence of nodes in the route to the node.
Set the distance value for the source node S to be 0.
Set the distance value for all other nodes to be INFINITY where this is to be set as a large value greater than any value that will be calculated.
While the ShortestPath set does not include all of the nodes do the following:
Pick the node (N) from the RemainingNodes set that has the lowest distance value.
Move this node into the ShortestPath set.
For each node in the RemainingNodes set that is adjacent to N:
Calculate a new distance value by adding the value given by the label of the edge connecting the two nodes to the already stored distance for N.
If this value is less than the value currently stored replace this stored value by the new one that has been calculated.
If a new value has been stored enter the sequence of nodes used to obtain this value.
Usage:
1. It is the basis of technology such as GPS tracking.
2. Google Maps
3. modelling the spread of infectious diseases
4. IP routing



Content of the ShortestPath set	Content of the record					
{}	A	B	C	D	E	F
	0	∞	∞	∞	∞	∞
{A}	A	B	C	D	E	F
	0	40	∞	∞	∞	60
	A	A-B				A-F
{A,B}	A	B	C	D	E	F
	0	40	50	∞	∞	55
	A	A-B	A-B-C			A-B-F
{A,B,C}	A	B	C	D	E	F
	0	40	50	90	∞	55
	A	A-B	A-B-C	A-B-C-D		A-B-F
{A,B,C,F}	A	B	C	D	E	F
	0	40	50	90	75	55
	A	A-B	A-B-C	A-B-C-D	A-B-F-E	A-B-F
{A,B,C,E,F}	A	B	C	D	E	F
	0	40	50	80	75	55
	A	A-B	A-B-C	A-B-F-E-D	A-B-F-E	A-B-F
{A,B,C,D,E,F}	A	B	C	D	E	F
	0	40	50	80	75	55
	A	A-B	A-B-C	A-B-F-E-D	A-B-F-E	A-B-F

Artificial neural network: based on the interconnections between neurons in the human brain. The system is able to think like a human using these neural networks, and its performance improves with more data.
Artificial neural networks are excellent at identifying patterns which would be too complex or time consuming for humans to carry out.
Back propagation of errors: An algorithm for machine learning that optimises the values for parameters which are adjustable. It is applied first to the nodes in the output layer and then works backward through the nodes in hidden layers until finally the input nodes are considered.
Deep Learning systems: structures algorithms in layers(input layout, out layout, hidden layout) to create an artificial neural network that can learn and make intelligent decisions on its own.

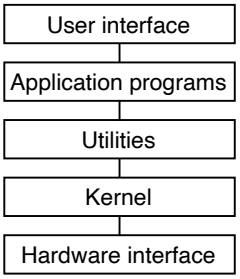
A* algorithm: find the best route from one node to just one other node A* algorithm is a modification of the Dijkstra algorithm designed to improve matters.
different between dijkstra and A*
 # Dijkstra
Calculate a new distance value by adding the value given by the label of the edge
connecting the two nodes to the already stored distance for N.
 # A*
Calculate a new distance value by adding the value given by the label of the edge
connecting the two nodes to the already stored distance for N.
Calculate an estimated value for the distance of N from the destination node and
add this to the new distance value.

Machine learning: where a system improves its performance through analysis of previous performance
1. a computer-based system has a defined task or tasks to perform
2. **knowledge is acquired** through the experience of performing the tasks
3. as a result of this experience and the knowledge gained the performance of future tasks is improved.
Unsupervised learning: where the machine learning takes place entirely through the system analysing and categorising the available data
1. In unsupervised learning the system has to draw its own conclusions from its experience of the results of the tasks it has performed.
2. Powerful computer systems having access to massive data banks are regularly used to make decisions based on previous actions recorded.
Unsupervised learning usage:
1. density estimation
2. k-mean clustering
Supervised learning: where sample data is supplied to the system with associated data relating to the outcome of its use.
1. require both input and output to be given to the model so ti can be trained.
2. The model uses labelled data, so the desired output for a given input is known.
2. Algorithms receive a set of inputs and the correct outputs to permit the learning process.
3. Once trained, the model is run using labelled data.
4. The results are compared with the expected output; if there are any errors, the model needs further refinement.
5. The model is run with unlabelled data to predict the outcome.
Supervised learning usage:
1. regression analysis
2. classification analysis
Reinforcement learning: where an agent learns by receiving graded rewards for actions taken
Reinforcement learning has some features similar to unsupervised learning and other features similar to supervised learning.
1. An agent is learning how best to perform in an environment.
2. The environment has many defined states.
3. At each step the agent takes an action.
4. An agent has a policy that guides its actions.
5. The policy is influenced by the recorded history and the knowledge of the current state of the environment.
6. An action changes the environment to a new state.
7. The agent receives a reward following an action which is a measure of how effective the action was in relation to the achievement of the overall goal.
8. The policy will guide the agent in deciding whether the next action should be exploiting knowledge already known or exploring a new avenue.
In summary, the aim is to maximise the reward values by improving the quality of the policy. It is a trial-and-error search for optimum performance. It requires many repeated attempts at the same problem.
Regression analysis: finding a mathematical function that provides the best fit to the actual outcomes when outcomes are calculated from previous inputs

ALevel CS C20 System Software (1)

© 2021,有鱼英语
https://collapsar.yuketang.net/
https://coding.yuketang.net/

aspects relating to the use of OS:
1. the operating system programs are **stored on disk** so there is no operating system.the computer has stored in ROM a **basic input/output system (BIOS)** which starts a bootstrap program. It is this bootstrap program that loads the **operating system into memory** and sets it running.
2. An operating system can provide facilities to have more than one program stored in memory.
3. The internal viewpoint concerns how the activities of the operating system are organised to best use the resources available. The external viewpoint concerns the facilities made available for the user of the system.
4. The major resources associated with the internal viewpoint are the CPU, the memory and the I/O system.



User mode: one available for the user or an application program.
Kernel mode: sole access to part of the memory and to certain system functions that user mode cannot access.

How os can maximise the use of resources:
Memory
1. Moving frequently accessed instructions to cache for faster recall as SRAM is used rather than DRAM for cache
2. Making use of virtual memory with paging or segmentation to swap memory to and from a disk
3. Partitioning memory dividing main memory into static/dynamic partitions to allow for more than one program/task to be available //multiprogramming
4, Removing unused items/tasks from RAM by marking a partition as available as soon as the process using it has terminated

Disk:
1. Disk caching a disk cache holds data that is frequently transferred to/from the disk the cache can be held on disk or in RAM
2. Compression utility decreasing the size of a file stored on disk in order fit more / larger files on the disk
3. Defragmentation utility files are rearranged to occupy contiguous disk space this reduces the time taken to access files// decreases latency

Memory management aspects:
1. The provision of protected memory space for the OS kernel.
2. The loading of a program into memory requires defining the memory addresses for the program itself, for associated procedures and for the data required by the program.
3. The storage of processes in main memory can get fragmented in the same way as happens for files stored on a hard disk.
4. Decisions have to be made about how large a part of memory should be allocated to individual processes sharing memory.

Partition: different processes were loaded into memory simultaneously need to partition memory
Dynamic partition: allowed to adjust to match the process size
Segmentation: where a large process is divided into segments for loading into memory but the segments are **not constrained to be the same size**
factors of limited the efficiency by segmentation:
1. the segments were not constrained to be the same size.
2. the size of process did not allow all of the segments for one process to be in memory at the same time.
3. These two factors combined to cause fragmentation both of the memory and of the disk storage. This resulted in degradation in the performance of the system.

Paging: where a large process is divided into pages which have to be **the same size**
1. The process is divided into equal-sized pages
2. memory is divided into frames of the same size
3. The secondary storage can also be divided into frames.
4. A large program is likely to have optional routes for the execution.

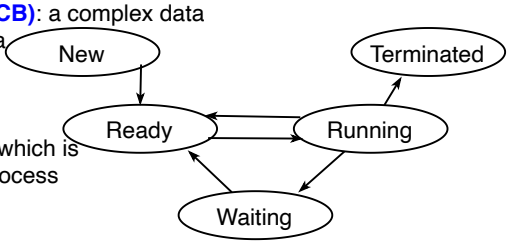
Virtual memory: a paging mechanism that allows a program to **use more memory addresses** than are available in main memory
An address has to comprise two parts: the **page number** plus the **offset** from the start of the page.
Page number: the page number **in logical memory**. The logic page number is **continuous**. The logical address stores the **page number in the four most significant bits** and the **page offset in the four least significant bits**.
Page frame number: the page number **in physical memory**. The page frames used **do not have to be adjacent**.
Presence flag: indicating whether or not the page is in memory.
Paging step:
1. the set of pages comprising a process are stored on disk.
2. One or more of these **pages is loaded into memory** when the process is changing to the ready state.
3. When the process is dispatched to the running state, the process starts executing.
4. the process will need access to a page that the page table indicates is **not in memory**. This is called a **page fault condition**.
5. in order to bring in the required page from secondary storage, **a page will need to be taken out of memory first.** use **first-in first-out method** or **least-recently-used method.**

Disk thrashing: when paging is being used and a repetitive state has been reached where loading one page causes a need for another page to be loaded almost immediately but the loading of this new page causes the same immediate need

Multitasking: allows computer to carry out multiple tasks at the same time.
1. Multitasking is actually multiple tasks processed at the same time, but actually processor can process one task at a time, so it has to swap between processes called scheduling.
2. Swapping happens so fast that it appears that all processes are running at the same time.
3. When there are too many processes, or some of them are making the CPU work especially hard, it can look as though some or all of them have stopped.
4. Multitasking doesn't mean that an unlimited number of tasks (process) can be juggled at the same time.
5. A process is a program that has started to be executed. A task that is to be executed or being executed by CPU is called process Each task consumes system storage and other resources. As more tasks are started, the system may slow down or begin to run out of shared storage.
6. Multitasking ensures the best use of computer resources by monitoring the state of each process.
7. The processes can be in running, ready or blocked state.
8. The Kernel of Operating system overlaps the execution of each process based on scheduling algorithm.

High-level scheduler: makes decisions about which program stored on disk should be moved into memory
Low-level scheduler: makes decisions about which process stored in memory should have access to the CPU
Process: a program in memory that has an associated process control block
Transitions between the status:
1. A new process arrives in memory and a **PCB is created**; it changes to the **ready state**.
2. A process in the ready state is **given access to the CPU** by the dispatcher; it changes to the **running state**.
3. A process in the running state is **halted by an interrupt**; it returns to the **ready state**.
4. A process in the running state cannot progress until **some event has occurred (I/O perhaps)**; it changes to the **waiting state** (sometimes called the 'suspended' or 'blocked' state).
5. A process in the waiting state is notified that **an event is completed**; it returns to the **ready state**.
6. A process in the running state **completes execution**; it changes to the **terminated state**.

Process control block (PCB): a complex data structure containing all data relevant to the running of a process
Thread
Thread: part of a program which is handled as an individual process when being executed
two reason for interrupts:



1. Processes consist of alternating periods of CPU usage and I/O usage. **I/O takes far too long for the CPU** to remain idle waiting for it to complete.
2. The **scheduler** decides to halt the process

Scheduling method:
preemptive: can halt a process that would otherwise continue running undisturbed.
non-preemptive:
Shortest job firs:
FCFS(first come first served):
1. **non-preemptive** algorithm
2. can be implemented by placing the process in a **first-in first-out(FIFO) queue**.
3. very **inefficient**

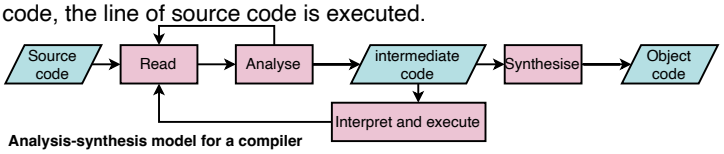
Round-robin:
1. **allocates a time slice** to each process
2. **preemptive** algorithm.
3. A process will be halted when its time slice has run out.
4. it can be implemented as a **FIFO queue**.

Priority-based scheduling Algorithm:
1. every time a new process enters the ready queue or when a running process is halted, the priorities for the processes may have to be re-evaluated.
2. **possible criteria**:
   1. estimated time of process execution
   2. estimated remaining time for execution
   3. length of time already spent in the ready queue
   4. whether the process is I/O bound or CPU bound.

ALevel CS C20 System Software (2)

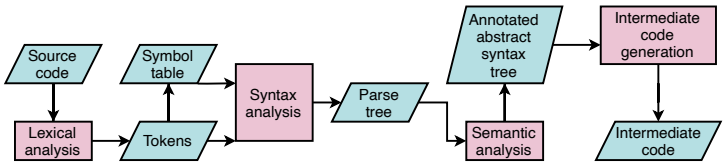
© 2021,有鱼英语
https://collapsar.yuketang.net/
https://coding.yuketang.net/

Backend program: takes this intermediate code as input and performs synthesis of object code.
Analyse -> Read: the source code is read line-by-line
intermediate code -> interpret and execute: once a line of source code has been found to be error free and therefore converted to intermediate code, the line of source code is executed.



Analysis-synthesis model for a compiler

Front-end program: performs analysis of the source code and unless errors are found produces an intermediate code that expresses completely the semantics (the meaning) of the source code.



Front-end analysis

Lexeme: Each meaningful individual character or collection of characters. A lexeme may be an identifier used by the programmer or may be a keyword, operator or symbol that is defined by the programming language.
lexical analysis: first to remove all white space and all comments then to take each line of source code and identify each lexeme.

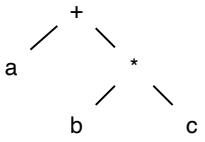
RPN:

1. An assignment statement often has an algebraic expression defining a new value for an identifier.
2. The expression can be evaluated by first converting the infix representation in the code to Reverse Polish Notation (RPN).

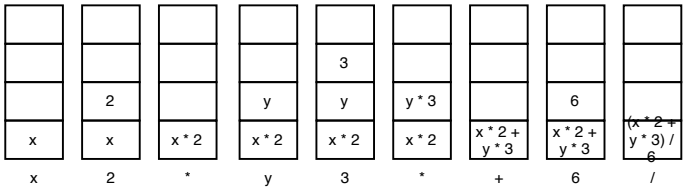
step:
1. Working from left to right in the expression
2. If element is a number PUSH that number onto the stack
3. If element is an operator then POP the first two numbers from stack
4. perform that operation on those numbers
5. PUSH result back onto stack
6. End once the last item in the expression has been dealt with

infix to RPN:
a + b * c --> a + b c * --> a b c * +

syntax tree



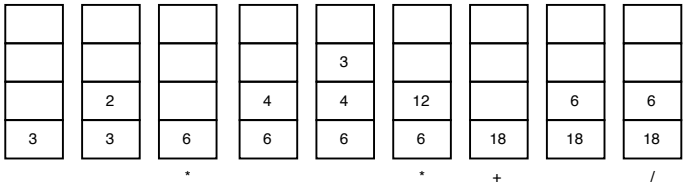
RPN to infix:
x 2 * y 3 * + 6 / --> (x * 2) y 3 * + 6 / --> (x * 2) (y * 3)+ 6 / -->
((x*2) + (y*3))6/ --> ((x*2) + (y*3)) / 6 --> (x * 2 + y * 3) / 6



(x * 2 + y * 3) / 6

Evaluating an RPN expression:

x 2 * y 3 * + 6 / (x = 3 ; y = 4)



Symbol table: a data structure in which each record contains the name and attributes of an identifier

1. For each identifier recognized there must be an entry made in the symbol table
2. contains identifier attributes such as the data type, where it is declared and where it is assigned a value.
3. The symbol table is an important data structure for a compiler.

```
Counter ← 0
INPUT Password
REPEAT
    IF Password <> "Cambridge"
        THEN
            INPUT Password
    ENDIF
    Counter ← Counter + 1
UNTIL Password = "Cambridge"
OUTPUT Counter
```

Keyword Token

Keyword	Token
←	01
+	02
+	02
IF	4A
THEN	4B
ENDIF	4C
ELSE	4D
REPEAT	4E
UNTIL	4F
TO	50
INPUT	51
OUTPUT	52
ENDFOR	53

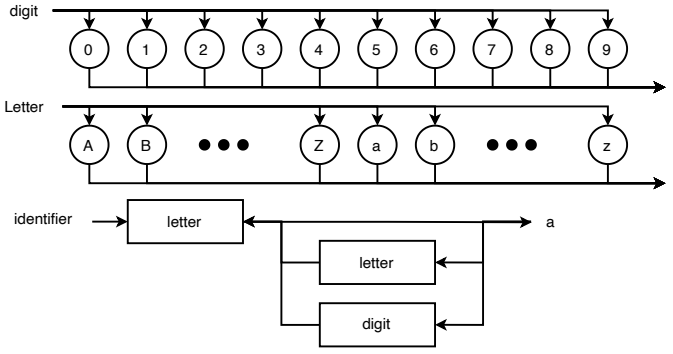
Symbol table

Symbol	Value(token)	Type(token)
Counter	60	Variable
0	61	Constant
Password	62	Variable
"Cambridge"	63	Constant
1	64	Constant

output from the lexical analysis

60	01	61	51	62	4E	4A	62
04	63	4B	51	62	4C	60	...

Syntax Diagram: defined the syntax allowed for an identifier



BNF(Naur Form notation): meta-language, used to describe the syntax and composition of statements which make up a high-level programming language.
1. programming languages
2. document formats
3. communication protocols

<Identifier> ::= <letter> | <identifier><letter>|<identifier><letter>
<Digit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
<Letter> ::= A | B | C | ... | Z | a | b | c | ... | z
<Letter> ::= <UpperCaseLetter>|<LowerCaseLetter>
<UpperCaseLetter> ::= A | B | C | ... | Z
<LowerCaseLetter> ::= a | b | c | ... | z
<binary digit> ::= 0 | 1

1. | is to separate individual options
2. ::= read as 'is defined as'
3. the recursive definition of <Identifier> in this particular version of BNF

Optimisation:
the main back-end stage is machine code generation from the intermediate code.
The aim of optimisation is to create an efficient program. One type of optimisation focuses on features that were inherent in the original source code and have been propagated into the intermediate code.

x := (a + b) * (a – b)
y := (a + 2 * b) * (a – b)

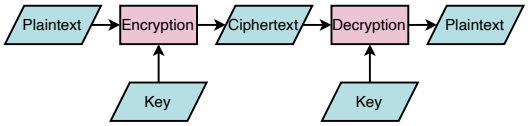
optimisation:
temp := (a – b)
x := (a + b) * temp
y := x + temp * b

1. Optimisation means that the code will have fewer instructions
2. Optimised code occupies less space in memory
3. Fewer instructions reduces the execution time of the program

ALevel CS C21 Security(1)

© 2021,有鱼英语
https://collapsar.yuketang.net/
https://coding.yuketang.net/

Plaintext: data before encryption
Ciphertext: the result of applying an encryption algorithm to data



Security concerns relating to a transmission:
1. **Confidentiality**: Only the intended recipient should be able to decrypt the ciphertext.
2. **Authenticity**: The receiver must be certain who sent the ciphertext.
3. **Integrity**: The ciphertext must not be modified during transmission.
4. **Non-repudiation**: Neither sender nor receiver should be able to deny involvement in the transmission.
5. **Availability**: Nothing should happen to prevent the receiver from receiving the transmission.

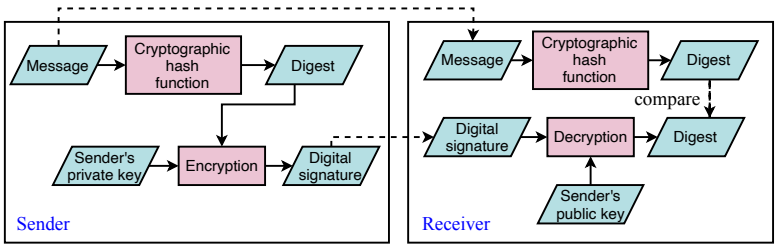
Symmetric key encryption: **one private key** is held by both sender and receiver and is used for both encryption and decryption
1. The issue with symmetric key encryption is delivery of the secret key.

Asymmetric key encryption: there is a **public key** and a **private key** one of which is used for encryption and the other for decryption
public key: sent to anyone who is going to partake in an encrypted communication.
private key: never sent to anyone.
The private and public keys are a matched pair.

Plaintext: data before encryption
Ciphertext: the result of applying an encryption algorithm to data
Asymmetric encryption:
An individual can encrypt a message with a **private key** and send this to a recipient who has the corresponding **public key** and who can then use this to decrypt the received ciphertext. **it could be used if it was important to verify who the sender was.** if the recipient finds that the decryption is successful, the message has in effect been received with a digital signature identifying the sender.
disadvantage: it is associated with an encryption of the whole of a message.
benefits:
1. Increased message security as one key is private
2. Allow message authentication
3. Allows non-repudiation
3. detects tempering

similarities between public key and private key:
1. both used in asymmetric
2. both use hashing algorithms
3. as a pair of key is required

differences between public key and private key:
1. private key only known to owner of the key pair, public key can be distributed to anyone
2. message encrypt with owner's public key, and only can decrypted by the owner's private key
3. digest are encrypted with the private key of the sender. and are encrypted with the public key of the receiver.



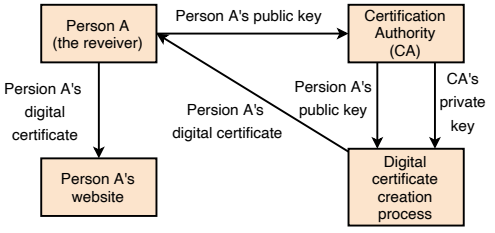
public cryptographic one-way hash function:
sender:
1. creates a number that is uniquely defined for the particular message, called a '**digest**'.
2. The private key is used to **encrypt the digest**.
3. The encrypted digest is the **digital signature**.
4. The message can be transmitted as **plaintext together with the encrypted digest as a separate file**.

receiver:
1. The same public one-way hash function is used to create a digest from the received message.
2. Then the encrypted version of the original digest is **decrypted using the public key.**
3. If the two digests are identical the receiver can be confident that the message is authentic and has been transmitted unaltered.

feature: the digital signature is different each time

CA digital certificate:
The receiver wants to be able to **receive secure messages from other individuals**, and these individuals want to be confident about the identity of the receiver. The public key must be made available in a way that ensures authentication
process of receiver to obtain a digital certificate to allow safe public key delivery:
1. An individual (person A) who is a would-be receiver and has a public –private key pair contacts a local CA.
2. The CA confirms the identity of person A.
3. Person A's public key is given to the CA.
4. The CA creates a public-key certificate (a digital certificate) and writes person A's public key into this document.
5. The CA uses encryption with the CA's private key to add a digital signature to this document. 6 The digital certificate is given to person A.
6. Person A posts the digital certificate on a website.

Receiver to obtain a digital certificate to allow safe public key delivery.



Digital certificate contains:
1. a hashing algorithm
2. a public key
3. serial number
4. dates valid
5. Name of subject
6. Name of issuer

Purpose of digital signature:
1. To ensure a document is authentic// came from a trusted source
2. To ensure a document has not been altered during transmission
3. Non repudiation

Digital signature step:
1. The message is hashed with the agreed hashing algorithm
2. ...to produce a message digest
3. The message digest is encrypted with the sender's private key
4. ... so the digital signature can be decrypted with sender's public key

similarities between digital certificate and digital signature:
1. both used for authentication
2. both are unique to the owner/subject
3. both use owner's public key
4. both make use of hash algorithm

differences between digital certificate and digital signature:
1. certificate obtained from issuing authority, signature created from a message
2. certificate provides authentication of owner, signature used to authenticate message that are sent by the owner
3. certificate remains unchanged whilst it is valid. new signature created for every message
4. only certificate provides extra information, only signature makes use of a private key

ALevel CS C21 Security(2)

© 2021,有鱼英语
https://collapsar.yuketang.net/
https://coding.yuketang.net/
有鱼英语

RSA key generation algorithm:
1. Two very large prime numbers p and q are chosen and their product n is calculated.
2. The product (p-1)(q-1) is calculated.
3. A prime number e less than (p-1)(q-1) and not a factor of it is chosen (65537 is the usual choice).
4. Another number d is found which satisfies the condition that the product of d times e when divided by (p-1)(q-1) gives a remainder of
5. The public key becomes the pair (n,e).
6 The private key becomes the pair (n,d).

TSL:
1. A protocol with two layers:
   1. **Record protocol**: can be used with or without encryption
   2. **Handshake protocol**: permits the website and the client to authenticate each other and to make use of encryption algorithm(a **secure session** between client and website is established)
2. A **TLS/digital/public key certificate** is used for authentication
3. Handshake use asymmetric cryptography
4. establish a **shared session key**
5. TLS can make use of **session caching** which improves the overall performance
6. one the session has been established. the client and server can agree which encryption algorithm are to be used and can define the values for the session keys that are to be used.

TSL usages:
1. Browsers accessing secure websites
2. VPNs
3. Email
4. VOIP

TSL purpose:
1. provide **secure communication**
2. maintain **data integrity**
3. additional **layer of security**

TSL applications:
1. online banking
2. private email
3. online shopping
4. online messages

handshake process:
1. the client sending some communication data to the server
2. the client asking the server to identify itself
3. the server sending its digital certificate including the public key.
4. The client validates (the server's) TLS Certificate
5. The client sends its digital certificate (to the server if requested)
6. Client sends an encrypted message to the server using the server's public key
7. The server can use its private key to decrypt the message • and get data needed for generating symmetric key
8. Both server and client compute symmetric key (to be used for encrypting messages) // session key established
9. The client sends back a digitally signed acknowledgement to start an encrypted session
10. The server sends back a digitally signed acknowledgement to start an encrypted session

Two concerns access a website:
1. whether or not the **website is genuine**
2. whether we can **transfer sensitive personal data to the website**

When the SSL protocol is in place, the application protocol HTTP becomes HTTPS.

SSL facts:
1. Although SSL is referred to as a protocol, it is in fact **a protocol suite.**
2. There is a **Record Protocol** that deals with the format for data transmission.
3. There is also a **Handshake Protocol responsible for security**.
4. The operation of SSL happens without any action from the user.
5. The **starting point for SSL** implementation is a connection between the client and the server being established by TCP.
6. The client browser then invokes the Handshake Protocol from the SSP suite.
7. The Handshake Protocol requests from the server its **SSL certificate** which is a digital certificate confirming its identity.
8. The server sends this SSL certificate plus its public key.
9. The browser uses this public key to encrypt a key which is to be used as a one-off session key for symmetric key encryption to be used for the data transfer during the session.
10. There may also be a need at this time to agree which encryption algorithms are to be used.

Security protocols: Secure Sockets Layer(SSL), Transport Layer Security(TSL)

SSL step:
1. the user's web browser sends a message so that it can connect with the required website which is secured by SSL
2. The web browser then requests that the web server identifies itself
3. The web server responds by sending a copy of its SSL certificate to the user's web browser
4. if the web browser can authenticate this certificate, it sends a message back to the web server to allow communication to begin.
5. Once this message is received, the web server acknowledges the web browser, and the SSL-encrypted two-way data transfer begins.