

Ch01

计算机系统 计算系统 计算机是一台设备。

计算系统是动态的.用于解决问题以及与其所处的环境进行交互。

计算系统由硬件、软件和它们管理的数据组成。

硬件：计算系统的物理元件（打印机、电路板、电线、键盘等）

软件：提供计算机执行的指令的程序

计算系统分层：通信层、应用层、操作系统层、程序设计层、硬件层、信息层（从外到内）

抽象：删除了复杂细节的心理模型

Ch02

二进制，八进制，十进制，十六进制的相互转化

Ch03

数据：基本值或者事实

信息：用有效的方式组织或处理过的数据。

计算机是一台多媒体设备，可以处理各种各样的信息。计算机可以存储、表示和帮助我们修改各种类型的数据 Numbers 数字

Text 文本

Audio 音频

Images and graphics 图像和图形

Video 视频

数据压缩：减少存储一段数据所需的空間。

压缩率：说明压缩的程度，是压缩后的数据大小除以原始数据大小的值。

数据压缩技术：

无损压缩，即提取的数据没有丢失任何原始信息

有损压缩，即在压缩过程中将丢失一些信息

负数表示法，二进制表示法：

符号数值表示法 [原码] 符号表示数所属的分类（正数或负数），值表示数的量值（解决正 0 和负 0）保持所有的数作为整数值，可以用一半数表示正数，另一半表示负数，

符号由数的量值决定

使用两个十进制数 让 1 至 49 表示正数 1 至 49，用 50 至 99 表示负数-50 至-1

在这种模式下执行加法，只需要对两个数求和，然后舍弃进位即可。

$A - B = A + (-B)$ 等价于给第一个数加上第二个数的负数

这种负数表示法称为十进制补码

$$\text{Negative}(I) = 10^k - I, \text{ where } k \text{ is the number of digits}$$

(k 为数字个数)

二进制补码

为了便于查看长的二进制数，我们把实数直线绘制成垂直的

二进制补码加法和减法同十进制补码运算法则一样

数字溢出 (overflow)

当我们分配给结果的位数存不下计算出的值时，将发生溢出

溢出是把无限的世界映射到有限的机器上会发生的典型问题。

实数表示法

实数具有整数部分和小数部分 小数点右侧的位置是十分位、百分位。

二进制实数表示法采用的是同十进制相同的规则 十进制数的点实际上相当于小数点
小数点右侧的位置是二分位、四分位、八分位。
十进制实数值可以用下列公式来表示[sign:符号, mantissa:尾数, exp:指数]

$$\text{sign} * \text{mantissa} * 10^{\text{exp}}$$

这种表示法称为浮点表示法, 因为数字的个数是固定的, 但是小数点却是浮动的。

下面的公式定义了一个二进制浮点值:

$\text{sign} * \text{mantissa} * 2^{\text{exp}}$ (次方)

文本表示法

要表示的字符数是有限的。一种表示字符的普通方法是列出所有字符, 然后赋予每个字符一个二进制字符串。

字符集只是字符和表示它们的代码的清单。计算机制造商赞同字符集标准化

ASCII 是美国信息交换标准代码 最初, ASCII 字符集采用 7 位每个字符, 可以表示 128 个不同的字符 之后, ASCII 字符集进化了, 用 8 位表示每个字符

扩展 ASCII 无法满足国际需要 一般情况下, 每个字符的编码使用 16 位 Unicode 被设计为 ASCII 的超集 前 256 个字符与扩展 ASCII 字符集中的完全一样

文本压缩:

在一个文档中, 每个字符分配 16 位, 可能占用太多的文件空间 我们需要更有效率的方法来存储和传输文本。 文本压缩技术

keyword encoding 关键字编码

run-length encoding 行程长度编码

Huffman encoding 赫夫曼编码

关键字编码: 它用单个字符代替常用的单词

行程长度编码: 一个字符可能在一个长序列中反复出现

赫夫曼编码 用变长的二进制串表示字符。使常用的字符具有较短的编码。

赫夫曼编码的一个重要特性是用于表示一个字符的位串不会是表示另一个字符的位串的前缀。 从左到右扫描一个位串时, 每当发现一个位串对应一个于一个字符, 那么这个位串就一定表示这个字符, 该位串不可能是更长位串的前缀。

音频数据表示法:

一个立体声系统通过把电信号发达到一个扬声器来制造声音。

这种信号是声波的模拟表示法。

信号中的电压按声波的正比例变化。

通过采样来数字化:

周期性地测量信号的电压 记录合适的值。

音频格式:

WAV, AU, AIFF, VQF, and MP3 MP3 (MPEG-2, audio layer 3 file) is dominant

图像与图形表示法

在计算机中, 颜色通常用 RGB (红-绿-蓝) 值表示, 这其实是三个数字, 说明了每种原色的相对份额。RGB 值 (255, 255, 0) 最大化了红色和绿色的份额, 最小化了蓝色的份额, 结果生成立了嫩黄色。

色深度: 用于表示颜色的数据量称为色深度, 通常用表示颜色的位数表示

(Hicolor) 增强彩色: 把色深度为 16 位的颜色, RGB 值中的每个数字由 5 位表示, 剩下的

一位有时用于表示透明度。

TrueColor 真彩色：指色深度为 24 位的颜色，RGB 值中的每个数字由 8 位表示。

数字化一幅图像：是把它表示为一个独立的点集，这些点称为像素。

分辨率：用于表示图像的像素个数。

光栅图形：逐个像素存储图像信息。

Bitmap (BMP), GIF, JPEG, and PNG are raster-graphics formats

图形的矢量表示法：

矢量图形：用线段和几何形表示图像的方法。

矢量图形是一系列描述线段的方法、线宽和颜色的命令。

由于不必记录所有的像素，所以采用这种格式的文件一般比较小。

矢量图形可以通过数学计算调整大小，这些改变可以根据需要动态地计算。

但是矢量图形不适用于表示真实世界的图像

视频表示法：

视频编译器：指用于缩减电影大小的方法，使电影能够在计算机或者网络上播放。

几乎所有的视频编译器都采用有损压缩，以最小化与视频相关的数据量。

Ch04

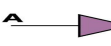
门：对电信号执行基本运算的设备，接受一个或多个输入信号，生成一个输出信号。


电路：是由门组合而成的，可以执行更加复杂的任务。


布尔表达式：使用布尔代数，用数学公式来表示两值逻辑

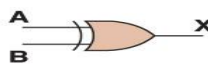
逻辑框图：是电路的图形化表示。每种类型的门由一个特定的图形符号表示。

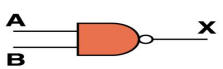
真值表：列出了一种门可能遇到的所有输入组合和相应的输出。


Boolean Expression	Logic Diagram Symbol	Truth Table						
$X = A'$		<table><tr><th>A</th><th>X</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	X	0	1	1	0
A	X							
0	1							
1	0							

Boolean Expression	Logic Diagram Symbol	Truth Table															
$X = A \cdot B$		<table><tr><th>A</th><th>B</th><th>X</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	X	0	0	0	0	1	0	1	0	0	1	1	1
A	B	X															
0	0	0															
0	1	0															
1	0	0															
1	1	1															

Boolean Expression	Logic Diagram Symbol	Truth Table															
$X = A + B$		<table><tr><th>A</th><th>B</th><th>X</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	X	0	0	0	0	1	1	1	0	1	1	1	1
A	B	X															
0	0	0															
0	1	1															
1	0	1															
1	1	1															

Boolean Expression	Logic Diagram Symbol	Truth Table															
$X = A \oplus B$		<table><tr><th>A</th><th>B</th><th>X</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	X	0	0	0	0	1	1	1	0	1	1	1	0
A	B	X															
0	0	0															
0	1	1															
1	0	1															
1	1	0															

Boolean Expression	Logic Diagram Symbol	Truth Table															
$X = (A \cdot B)'$		<table><tr><th>A</th><th>B</th><th>X</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	X	0	0	1	0	1	1	1	0	1	1	1	0
A	B	X															
0	0	1															
0	1	1															
1	0	1															
1	1	0															

Boolean Expression	Logic Diagram Symbol	Truth Table															
$X = (A + B)'$		<table border="1"> <thead> <tr> <th>A</th><th>B</th><th>X</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	A	B	X	0	0	1	0	1	0	1	0	0	1	1	0
A	B	X															
0	0	1															
0	1	0															
1	0	0															
1	1	0															

组合电路:输出仅由输入值决定的电路

时序电路: 输出是输入值和电路当前状态的函数的电路

组合电路: 把一个门的输出作为另一个门的输入, 就可以把门组合成电路

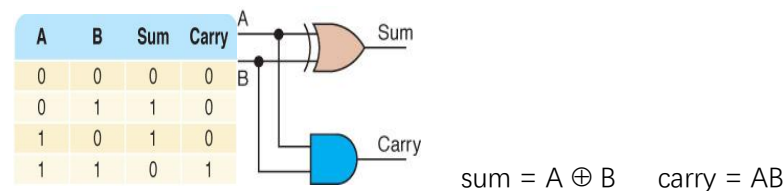
电路等价: 对应每个输入值组合, 两个电路都生成完全相同的输出

布尔代数的一大优点, 它们利用可以证明的数学法则来设计逻辑电路

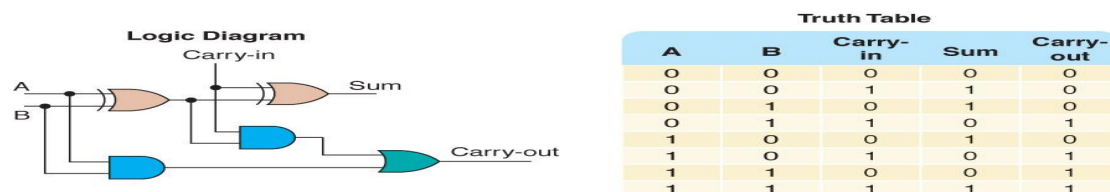
在数字逻辑层, 加法是用二进制执行的。

这些加法运算是由专用电路加法器执行的。

半加器: 计算两个数位的和并生成正确进位的电路。



全加器: 计算两个数位的和, 并考虑进位输入的电路



Ch06

计算机是能够存储、检索和处理数据的可编程电子设备

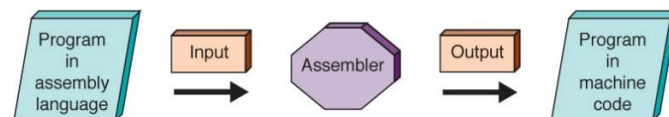
数据和操作数据的指令逻辑上是相同的, 它们存储在相同的地方。

机器语言: 由计算机直接使用的二进制编码指令构成的语言。

汇编语言: 一种低级语言, 用助记码表示特定计算机的机器语言指令。

汇编器: 把汇编语言程序翻译成机器代码的程序。

汇编过程:



算法: 解决方案的计划或概要, 或解决问题的逻辑步骤顺序.

伪代码中没有语法规则, 不考虑大小写

Ch07

问题求解: 为一个令人困惑、苦恼、烦恼或未决的问题找到解决办法的行为。

如何解决它这个列表是普通适用的。虽然是在解决数学问题这个背景下编写的 我们可以用来解决与计算机相关问题。

如何解决问题:

Understand the problem 理解问题

Devise a plan 制订一个方案

Carry out the plan 执行方案

Look back 分析得到的解决方案

计算机解决问题过程

Analysis and Specification Phase 分析和说明阶段

Analyze 分析 【理解（定义）问题】

Specification 说明 【说明程序要解决的问题】

Algorithm Development Phase 算法开发阶段

Develop algorithm 开发算法 【开发用于解决问题的逻辑步骤序列】

Test algorithm 测试算法 【执行列出的步骤，看它们是否能真正地解决问题】

Implementation Phase 实现阶段

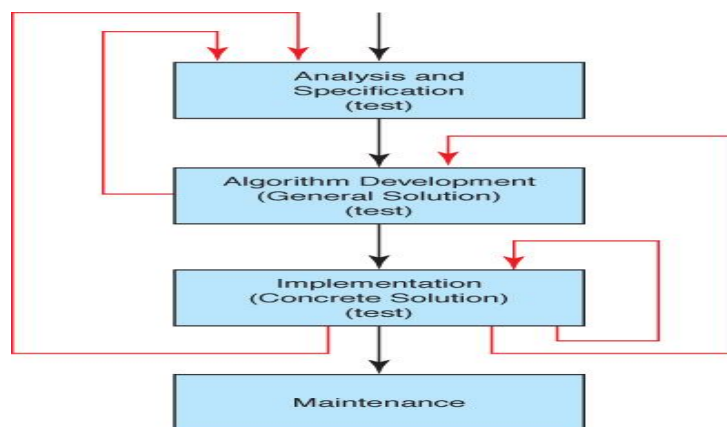
Code 编码 【用程序设计语言翻译算法（通用解决方案）】

Test 测试 【让计算机执行指令序列。检查结果，修改程序，直到得到正确的答案】

Maintenance Phase 维护阶段

Use 使用 【使用程序】

Maintain 维护 【修改程序，使它满足改变了的要求，或者纠正其中的错误】



方法总结：

Analyze the Problem 分析问题

Understand the problem!! 理解问题

Develop a plan of attack 开发一个通用的方案

List the Main Tasks (becomes Main Module) 列出主要任务（成为主要模块）

Restate problem as a list of tasks (modules) 以任务列表（模块）的方式重新描述问题

Give each task a name 给每个任务命名

Write the Remaining Modules 编写其余模块

Restate each abstract module as a list of tasks 重申每个抽象模块作为一个任务列表

Give each task a name 给每个任务命名

Re-sequence and Revise as Necessary 根据需要进行重组和改写

Process ends when all steps (modules) are concrete 过程结束时，所有的步骤（模块）是具体的

算法：解决问题的逻辑步骤序列

在有限的时间内用有限的数据解决问题或者子问题的明确指令集合

抽象步骤：含有不明细节的算法步骤 具体步骤

开发算法 计算机解决问题方法：

自顶向下设计聚焦要完成的任务

面向对象设计着重于解决方案中涉及的数据

计数控制循环可以指定过程重复的次数,这个循环的机制是简单记录过程重复的次数,且在重复再次开始前检测循环是否已经结束。

事件控制循环是由循环体自身内发生的事件控制的循环。

复合数据类型

记录:是异构项目的有名集合,可以通过名字单独访问其中的项目

数组:是同构项目的有名集合,可以通过单个项目在集合中的位置访问它们.项目在集合中的位置叫做索引.

无序数组中顺序搜索:

顺序搜索依次查找每一个元素并将其与我们需要的元素进行比较.

如果匹配,则找到了这个元素.如果不匹配,则继续找下一个元素.

当我们发现了元素或者是查找所有的元素后都没有找到匹配项就停止.

这听起来象是一个有两个结束条件的循环

顺序搜索算法

布尔运算

有序数组:所谓排序,就是使一串记录,按照其中的某个或某些关键字的大小,递增或递减的排列起来的操作。

有序数组中顺序搜索: 如果知道数组中的项目是有序的,那么在查找时,如果我们所需要的项目在数组中,到了这个数可能在数组中的位置时就可以停止查找了.

二分检索:

顺序检索:在数组开头开始,直到找到匹配项或者整个数组中都没有匹配项.

二分检索(要检索的数组是有序的):是从数组中间开始,之后决定你要查找的单词在左手部分或者是右手部分,找到正确的部分后再重复这个过程。

排序 排序算法 选择排序 冒泡排序 插入排序

当数组在排序之前有序时, 冒泡排序算法较好

子程序语句: 我们可以给一段代码一个名称, 然后程序另一部分的一个语句使用这个名称。

遇到这个名称时, 这个进程的其他部分将会终止, 等待这个命名代码被执行。

形式参数 实际参数

递归:

递归: 算法调用它本身的能力

基本情况: 是答案已知的情况

一般情况: 是调用自身来解决问题的更小版本的解决方案

Ch08

抽象数据类型:属性(数据与操作)明确地与特定实现分离的数据类型.

设计的目标是通过抽象减少复杂度.

复合数据类型

数据结构:一种抽象数据类型中的复合数据域的实现

容器:存放和操作其他对象的对象

容器的两种逻辑实现

基于数组的实现 基于链式的实现

堆栈是一种抽象复合结构,只能从一端访问堆栈中的元素 LIFO Push Pop

队列:队列是一种抽象结构,队列中的项目从一端入,从另一端出.FIFO

列表

Add item Put an item into the list

Remove item Remove an item from the list

Get next item Get (look) at the next item

more items Are there more items?

从文件中读取值,将值放到列表中,之后输出列表

树:

像列表,堆栈和队列这样的抽象结构本质上都是线性的,只模拟了一种数据关系.

更复杂的关系需要更复杂的结构来表示.

二叉树:具有唯一起始节点(根节点)的抽象复合结构,其中每个节点可以有二个子女节点,根节点和每个节点之间都有具有一条路径.

二叉检索树 BST:二叉检索树具有二叉树的形状属性,也就是说,二叉检索树中的节点可以具有 0 个,1 个或者 2 个子女.此外,二叉检索树还具有语义属性来刻画树中节点的值,即任何节点的值都要大于它的左子树中的所有节点的值,并且要小于它的右子树中的所有节点的值.

图:由一组节点和一组把节点相互连接起来的边构成的数据结构

无向图 :其中的边没有方向的图

有向图:其中的边是从一个顶点指向另一个顶点(或同一个顶点)的图

Ch09

面向对象的设计方法是用叫做对象的独立实体生成解决方案的问题求解方法.

对象:在问题背景中相关的事物或实体

对象类或者类:一组具有相似的属性和行为的对象的描述

方法:定义了类的一种行为的特定算法

类包含 Properties 属性 behaviors 行为(responsibilities 责任)

自顶向下设计 面向对象设计 任务 对象合作

面向对象

头脑风暴找到可能的类[与此问题相关的事物或者实体] 过滤重复的类或删除不必要的类
场景 责任算法

编译器:把用高级语言编写的程序翻译成机器码的程序

解释器:输入用高级语言编写的程序,指导计算机执行每个语句指定的动作的程序

解释器在翻译过程语句之后会立即执行这个语句

虚拟机

Java

Java 虚拟机 字节码 可移植性

可移植性: 一个程序运行在不同的机器上的能力

编译器(字节码)的可移植性

面向对象语言的功能性

封装:实施信息隐藏的语言特性 继承:类获取其他类的属性(数据域和方法)的机制

多态:一种语言的继承体系结构中具有两个同名方法且能够根据对象应用合适的方法的能力

Ch10

应用软件: 是为了满足特定需要——解决现实世界中的问题——而编写的

系统软件: 负责在基础层上管理计算机系统

操作系统是最重要的系统软件

负责管理计算机的资源, 如内存和输入输出设备

提供人机交互的界面

允许一个应用程序与其他系统资源

操作系统的各种角色通常都围绕着一个中心思想“良好的共享”

操作系统负责管理资源，而这些资源通常是由使用它们的程序共享的
资源管理

多道程序设计：同时主存中驻留多个程序，由它们竞争 CPU 的技术

内存管理：了解主存中载有多少个程序以及它们的动作

进程：程序正在执行过程中的动态表示法

进程管理：跟踪进程的进展以及所有中间状态

CPU 调度：确定主存中的哪个进程可以访问 CPU 以便执行的动作

操作员会把来自多个用户的作业组织成分批。一个分批包含一组需要相同或相似资源的作业。

分时共享系统：允许多个用户同时与计算机进行交互。

虚拟机：分时共享系统创建的每个用户都有专有机器的假象

实时系统：应用程序的特性决定了响应时间至关重要的系统

响应时间：收到信号和生成响应之间的延迟时间

设备驱动程序：针对特定设备的一个小程序，它负责这个设备的信息接收和发送

内存管理

跟踪一个程序驻留在内存的什么位置以及如何驻留的 把逻辑程序地址转换成实际的内存地址 逻辑地址：对一个存储值的引用，是相对于引用它的程序的

物理地址：主存储设备中的真实地址

内存中只有两个程序——操作系统和要执行的程序。这种方法称为单块内存管理

在这种内存管理机制中，逻辑地址只是一个相对于程序起始位置的整数值。要生成物理地址，只要用逻辑地址加上程序在存中的起始地址就可以了。

分区内存管理内存中只同时存在操作系统和一个其他的应用程序

分区内存管理中有操作系统以及任意数目的其他程序同时驻留在内存中

固定分区法：把内存分成特定数目的分区以载入程序的内存管理方法

动态分区法：根据容纳程序的需要对内存分区的内存管理方法

内存被划分为一组分区，有些是空的，有些分配给了程序

基址寄存器：存放当前分区的起始地址的寄存器

界限寄存器：存放当前分区的长度的寄存器

分区选择算法

最先匹配：即把第一个足够容纳程序的分区分配给它

最佳匹配：即把最小的能够容纳程序的分区分配给它

最差匹配：即把最大的能够容纳的分区分配给它

页式内存管理

页式内存技术：主存成小的大小固定的存储块称为帧的技术

帧：大小固定的一部分主存，用于存放进程页。

页：大小固定的一部分进程，存储在内存帧中

页编号：用页面大小除逻辑地址得到的商

偏移值：用页面大小除逻辑地址得到的余数

页映射表 PMP：操作系统用于记录页和帧之间的关系的表

请求分页：页式内存管理法的扩展，只有当页面被引用（请求）时才会被载入内存

页面交换：把一个页面从二级存储设备载入内存，通常会使另一个页面从内存中删除

虚拟内存：由于整个程序不必同时处于内存而造成的小没有限制的假象

系统颠簸：频繁的页面交换造成的低效处理

进程管理：单个进程使用 CPU 的管理动作

进程状态

PCB 进程控制块：操作系统管理进程信息使用的数据结构，包括
程序计数器的当前值 进程在其他所有 CPU 寄存器中的值 基址寄存器和界限寄存器的值
(或页表) 帐户信息

每个状态由一个 PCB 列表表示，处于该状态的每个进程对应一个 PCB

只有一个 CPU，因此只有一套 CPU 寄存器。

这些寄存器存放的是当前执行的进程的值。

每当一个进程进入了运行状态：

当前正在运行进程的寄存器值将被存入它的 PCB

它对应的 PCB 也会从一个状态列表中转移到另一个状态列表

新运行的进程的寄存器值将被载入 CPU

这种信息交换叫作上下文切换

CPU 调度：确定把哪个处于准备就绪状态的进程移入运行状态

可能有很多进程同时处于准备就绪状态

在任意一个时刻，只有一个进程可以进入运行状态

非抢先调度:当当前执行的进程自愿放弃了 CPU 时发生的 CPU 调度

抢先调度:当操作系统决定照顾别一个进程而抢占当前执行进程的 CPU 资源时发生的 CPU 调度

周转时间:从进程进入准备就绪状态到它最终完成之间的时间间隔,是评估 CPU 调度算法的标准

CPU 调度算法

先到先服务:进程按照它们到达运行状态的顺序转移到 CPU

最短作业优先:CPU 调度算法将查看所有处于准备就绪状态的进程,并分派一个具有最短服务时间的进程

轮询法:将把处理时间平均分配给所有准备就绪的进程

每个进程同等对待

时间片(量):在每个进程被抢占并返回准备就绪状态之前收到的时间量.

Ch11

文件：数据的有名集合，用于组织二级存储设备

文件系统：操作系统为它管理的文件提供的逻辑视图

目录：文件的有名分组

文本文件：数据字节是 ASCII 或 Unicode 字符集中的字符

二进制文件：包含特定格式的数据的文件，要求给位串一个特定的解释

文件类型：文件中存放的关于类型的信息。

大多数文件都包含有特定类型的信息

文件扩展名：文件名中说明文件类型的部分

文件名通常分为二个部分 文件名.文件扩展名

文件执行的操作

创建文件/删除文件

打开文件/关闭文件

从文件中读取数据/把数据写入文件

删减文件(删除它的内容)

重定位当前文件指针/把数据附加到文件结尾

重新命名文件/复制文件

文件访问

顺序文件：以线性方式访问文件中数据的方法。为了得到最后的纪录，你必须读完所有的记录

直接访问：文件中的数据会被概念性地划分为带编号的逻辑记录，通过指定逻辑记录编号直接访问的数据的方法

文件保护：限制文件访问的过程

在多用户系统中，文件保护的重要性居于首要地位

除非是特许的，否则我们不想让一个用户访问另一个用户的文件

文件保护机制都决定了谁可以使用文件，以及为什么目的而使用文件

目录是文件的有名集合。一个目录还可以包含另一个目录

父目录：包含其他目录的目录

子目录：被包含的目录

目录树：展示文件系统的嵌套目录组织的结构

根目录：最高级的目录

无论何时，你都可以认为自己在文件系统上的某个特定位置（即特定的子目录）工作

工作目录：当前活动的子目录

路径：文件或子目录在文件系统上的位置的文本名称

绝对路径：从根目录开始，包括所有后继子目录的目录

相对路径：从当前工作目录开始的路径

磁盘调度：操作系统用于决定先满足哪个请求的方法

在计算机同时处理多个中时，将建立一个访问磁盘的请求列表

FCFS 先到先服务：按照请求到达的顺序处理它们，并不考虑读写头的当前位置

SSTF 最短寻道时间优先磁盘调度法：将通过尽可能少的读写头移动满足所有未解决的请求

SCAN 磁盘调度算法的工作方式与之类似，只是在磁盘调度算法中没有上下移动，而是读写头向轴心移动，然后再向盘片边缘移动，就这样在轴心和盘片边缘之间来回移动。

Ch12

信息系统：帮助我们组织和分析数据的软件

信息管理：

电子表格和数据库管理系统：允许用户用不同的方式来组织、管理和分析数据的软件工具。

电子制表软件：是一种软件应用程序，它允许用户用带标签的单元格组织和分析数据

电子数据表中用于存放数据或公式的元素。

存储在单元格中的数据既可以是文本，也可以是其他特殊的数据(如日期)。

可以用行列标号引用电子数据表格中的单元格

电子表格公式

电子数据表的这种能力源于我们创建并存储在单元格中的公式】

把公式存储在一个单元格中，这个单元格就会显示该公式的结果

电子数据表函数：电子制表软件提供的可用于公式的计算函数。

范围：用端点指定的一组连续单元格

数据库：结构化的数据集合。

数据库管理系统：由物理数据库、数据库引擎和数据库模式构成的软件和数据的组合

物理数据库：存放数据的文件的集合

数据库引擎：支持对数据库内容的访问和修改的软件

数据库模式：存储在数据库中的数据的逻辑结构的规约。

数据库查询：从数据库检索数据的请求。

关系型数据库管理系统：用表组织数据和数据之间的关系

表：数据库纪录的集合

记录（对象，实体）：构成一个数据库实体的相关的域的集合。

域（属性）：数据库记录中的一个值。

键（关键字）：在表的所有记录中唯一标识一个数据库记录的一个或多个域关系

SQL 结构化查询语言：用于管理和查询数据的综合性关系数据库语言

修改数据库内容

数据库设计：。

实体关系 ER 建模：设计关系数据库的常用方法。

ER 图：ER 图形化表示

基数约束：在 ER 图中，一次可以存在于实体间折关系数量

Ch13

人工智能:研究对人类思想建模和应用人类智能的计算机系统的学科

例如,写一个程序在一幅图片中识别一个物体

图灵测试:根据经验来判断一台计算机是否达到了智能化

弱等价性:两个系统基于其结果的等价性,但实现这种结果的方式不同.

强等价性:两个系统基于其结果和实现这种结果的处理方法的等价性.

罗布纳奖:正式的图灵测试.每年举行一次.

聊天机器人:用于执行人机对话的程序.

Aspects of AI 人工智能问题的各个方面

Knowledge representation 知识表示

Expert systems 专家系统

Neural networks 神经网络

Natural language processing 自然语言处理

Robotics 机器人学

知识表示

我们想独立于数据的底层实现来创建它的逻辑视图,以便能用特定的方式处理数据

自然语言的说明性很强,但它不容易处理

语义网和检索树是很有前途的知识表示技术

语义网:是一种知识表示法,重点在于对象之间的关系.

表示语义网的是有向图

检索树:表示对抗性情况(如博弈)中的所有选择的结构

在检索树中,一条路径表示玩家做出的一系列决定

专家系统

基于知识的系统:使用特定信息集合的软件,可以从中提取和处理特定的片段

专家系统:基于人类专家的知识的软件系统

它是一个

基于规则的系统:基于一套 if-then 规则的软件系统

推理机:处理规则以得出结论的软件

人工神经网络:尝试模拟人体神经网络的计算机知识表示法

神经网络路径中的流量是稳定的

在我们学习新事物时,大脑中将构成新强神经路径

人工神经网络中的每个处理元素都类似于一个生物神经元

一个元素接受一定数量的输入值,生成一个输出值 0 或者 1

每个输入值有一个数字权

元素的有效权是所有输入值与它的权的乘积之和

$$v1 * w1 + v2 * w2 + v3 * w3$$

每个元素都有一个数字阈值

如果有效权大于阈值,这个元素将生成 1

如果有效权小于等于阈值,该元素将生成 0

训练: 调整神经网络中的权和阈值以实现想要的结果的过程

训练一个神经网络去找出一幅相片中的猫: 每个像素对应一个输出值,训练网络来对属于猫的每一个像素产生值为 1,对每一个不属于猫的像素产生值为 0

自然语言处理:

人机语音交互过程中的三种基本处理类型

语音合成:用计算机制造出人类的语言

语音识别:用计算机来识别人类所讲的话

自然语言理解:用计算机对人类传达的信息做出合理的解释

词法二义性: 由于单词具有多种含义而造成的二义性

句法二义性: 由于句子的构造方式有多种而造成的二义性

指代二义性: 由于代词可以指代多个对象而造成的二义性

Ch15

计算机网络: 是为了通信和共享资源而以各种方式连接在一起的一组计算设备

计算设备之间的连接可以使用物理电线或电缆, 也可以使用无线电波或红外信号

计算机网络开创了一个新的计算领域——客户/服务器模型

协议: 定义如何在网络上格式化和处理数据的一组规则。

LAN 局域网: 连接较小地理的少量计算机的网络

环形拓扑: 把所有节点连接成一个封闭的环, 消息在环中沿着一个方向传播。

星形拓扑: 以一个节点为中心, 其他节点都中心节点上, 所有消息都经过中心节点发送。

总线拓扑: 所有节点都一条通信线上, 消息可以在上双向传播。

被称为以太网的总线技术已经成为局域网的业界标准

WAN 广域网: 是连接两个或者多个相距较远的局域网的网络

MAN 城域网: 为大城市开发的网络基础设施

网关: 处理它的 LAN 和其他网络之间通信的节点

互联网: 跨越地球的广域网

包交换:

包:在网路上传输的数据单位

路由器: 指导包在网路上向最终目的地传输的网络设备

消息被分割为大小固定,有编号的包.每个包将独立在网上传输,直到到达目的地,它们将在此被重新组合为原始的消息. 这种方法叫作包交换.

开放系统 一个逻辑进步

专有系统 互通性 开放系统

应用层

表示层

会话层

传输层

网络层

数据链路层

物理层

网络协议

每一层都能依靠自己的基础协议

有时叫作协议栈

TCP 传输控制协议: 把消息分割成包,在目的地把包重新组装成消息,并负责处理错误的网络协议.

IP 网际协议: 处理包通过互相连接的网络传送到最终目的地的路由选择.

UDP 用户数据报协议 UDP: 牺牲一定可靠性实现较高传输速率的网络协议,是 TCP 的替代者

Ping: 用于测试一台特定的网络计算机量否是活动的以及是否可到达的程序.

跟踪路由程序:用于展示包在到达目的节点的过程中经过的路线的程序.

高层协议

Simple Mail Transfer Protocol (SMTP) File Transfer Protocol (FTP) Telnet Hyper Text Transfer Protocol (http)

防火墙

一台网关机器,它的软件通过过滤网络通信来保护网络

访问控制策略: 一个组织建立的一组规则,规定了接受和拒绝什么类型的网络通信.

云计算: 提供存储空间和互联网上其他资源的服务.

公有云:允许任何订阅用户访问

私有云:专门为一个特定的组或组织建立,并限制在该组织团体内访问.

社区云:在拥有相似需求的多个组织之间进行云共享

混合云:上述云服务类型的一些组合.

软件工程概论 Ch01

System 相互关联的 互相依赖的 协作的 software 预先决定 software engineering

软件生命周期 软件开发生命周期

SDLC Phase	Related Deliverable(s)
Investigation and Analysis	Initial System Requirements; Requirements Specification
Design (Modeling)	Design Specification
Development (Construction)	Actual Software System; Product Documentation
Implementation	
Management	Enhanced Software System; Revised Documentation

- Waterfall Model 瀑布模型
- Phased Prototype Model 阶段性原型模型
- Iterative Development Model 迭代开发模型
- Rapid Prototype Model 快速原型模型

- Agile Development Model 敏捷开发模型

计算机软件未来前景

Maintainability 可维护性: How easily maintained is the software?

This will depend on the quality of the design as well as the documentation.

- Documentation 文档: How well documented is the system?
- Efficiency 效率: How efficiently are certain core operations carried out? Of importance are the response time and the system throughput.

User Friendliness 用户友好性: How well designed is the user interface? How easy is it to learn and use the system?

- Compatibility 兼容性: with existing software products.
- Security 安全性: Are there mechanisms to protect and promote confidentiality and proper access control?
- Integrity 完整一致性: What is the quality of data validation methods?
- Reliability 可靠性: Will the software perform according to requirements at all times?
- Growth potential(增长潜力,可扩展性): What is the storage capacity? What is the capacity for growth in data storage?

Functionality and Flexibility(功能性和灵活性): Does the software provide the essential functional features required for the problem domain? Are there alternate ways of doing things?

- Differentiation(差异化): What is unique about the software?
- Adaptability(适应性): How well are unanticipated situations handled?
- Productivity(生产力): How will productivity be affected by the software system?
- Comprehensive Coverage (综合保障): Is the problem comprehensively and appropriately addressed?

软件质量

Real Play

When I taught the introduction-to-theater course at North Dakota State University, I required my students to attend the university theater's current production and write a **critique**. After viewing a particularly fine performance, one student wrote: "The play was so real, I thought I was actually sitting on my **couch** at home, watching it on television."

逼真的戏剧

我在北达科他州立大学教戏剧入门课时，要求学生们去看学校剧团当时的演出，并写一篇评论。看了一场极为精彩的演出后，一名学生写道：“这部戏剧是如此逼真，以致于我认为我自己是坐在家里的沙发上，从电视上看到的。”

学习资料 就找包打听

资料获取，回复公众号资料关键词

华工小朋友

包包！公众号我发了口令，
但是没有受到资料诶？



包包

要输入正确的口令才行噢，可以用盲猜法
(课程+试卷) 或者资料专区检索 (详见P4)

华工小朋友

如果口令、链接失效或者公众号
没有找到想要的资料，怎么办呢？



包包

别急，包包是人工运营的，
你可以通过以下途径反馈~ (P3)

包包有偿收集资料投稿

还有疑问？
找包子妹妹！



华工包打听公众号



包子妹妹



资料反馈箱



资料获取指南

华工包打听



资料声明

关于资料

· 来源

由同学投稿，包打听有偿收集、整理。

· 分享

资料无偿分享给同学使用

注意事项

资料不保证100%正确，仅供参考，切勿依赖
资料如有错误，请反馈给包打听微信
未经授权不能转作他用

华工新生答疑、校园指引、入学考试、感情树洞、华工黑市群、学习群、闲置群、校园资讯、校内通知、吃喝玩乐、兼职、家教、大学学车、考研、留学四六级（星球包）等一站式服务。

最全能校园
服务平台
校园大小事
皆可打听

华工包打听



包子妹妹



微信号——即时互动，
丰富社群，校园生活
资讯。

公众号——学习资料，
校园百事，学校通
知。

包星球——吃喝玩乐，
兼职考研留学信息，应
有尽有

QQ号——空间动态，
百事打听！