

16 System Software 系统软件

16.1 Purposes of an Operating System 操作系统的目的

Multi-tasking 多任务

Managing the execution of many programs that appear to run at the same time.

Scheduling 调度

Managing the processes running on the CPU.

process states 进程状态

the states of a process requiring execution the scheduler refers to when making decisions. At any given moment, a process can be either ready (waiting for CPU time), running (currently using the CPU), or blocked (neither using nor waiting to use the CPU, such as waiting for an event).

virtual memory 虚拟内存

Using secondary storage to temporarily simulate additional main memory so the CPU appears to be able to access more memory space than the available RAM. The data not currently in use are swapped from the main memory (RAM) to the virtual memory.

Paging 内存分页

Reading/writing same-size blocks of data from/to secondary storage when required.

Segmentation 内存分段

Locating non-contiguous blocks of data and relocating them.

Interrupt handling 中断处理

Transferring control to another routine when a service is required.

16.2 Translation Software 程序语言处理软件**front-end analysis 前端分析**

the first part of compilation, aiming at establishing the meaning of the source code. It consists of lexical analysis, syntax analysis, semantic analysis and intermediate code generation, in that order.

back-end synthesis 后端合成

the second part of compilation that generates the object (machine) code from the intermediate code. Optimisation is usually done before the machine code is produced.

lexical analysis 词法分析

the first stage of compilation. The comments and whitespaces are removed. Then the source code (as a sequence of characters) is identified via pattern-matching as a sequence of lexemes, each of which correspond to a token.

symbol table 符号表

a data structure in which each record contains the name and attribute of an identifier. It is generated during the lexical analysis.

syntax analysis 语法分析

the second stage of compilation, also known as **parsing**. Firstly, the list of tokens is checked for syntax error. If there is none, a parsing algorithms is used to interpret token list and produce a syntax tree (parse tree).

semantic analysis 语义分析

the third stage of compilation that is about establishing the full meaning of the code. It constructs an annotated abstract syntax tree.

intermediate code generation 中间代码生成

the fourth stage of compilation that converts the abstract syntax tree into an intermediate code.

optimisation 优化

a stage of compilation aiming at minimising a program' s execution time and memory requirement.

code generation 机器码生成

the last stage of compilation. converting the optimised intermediate code into an executable form.