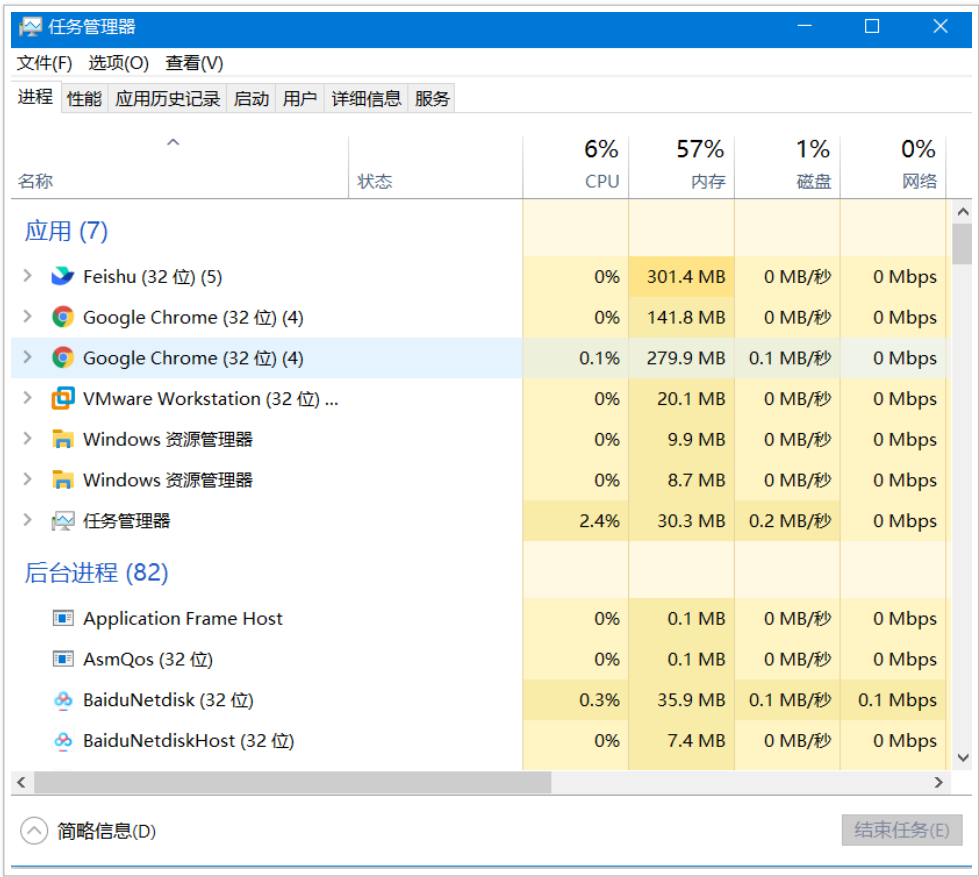


1.1 进程的概念_物联网 / 嵌入式工程师 - 慕课网

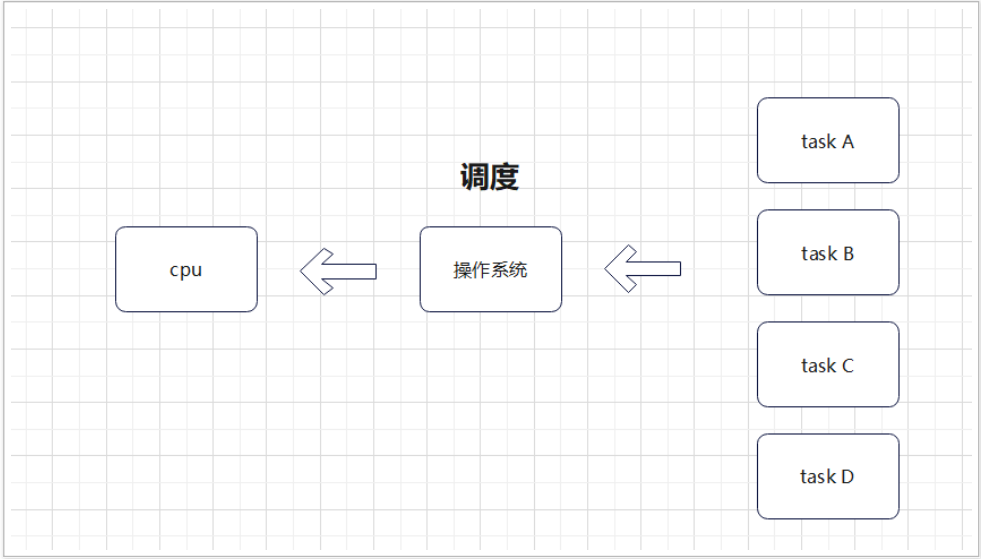
“ 慕课网慕课教程 1.1 进程的概念涵盖海量编程基础技术教程，以图文图表的形式，把晦涩难懂的编程专业用语，以通俗易懂的方式呈现给用户。

1. 进程的概念

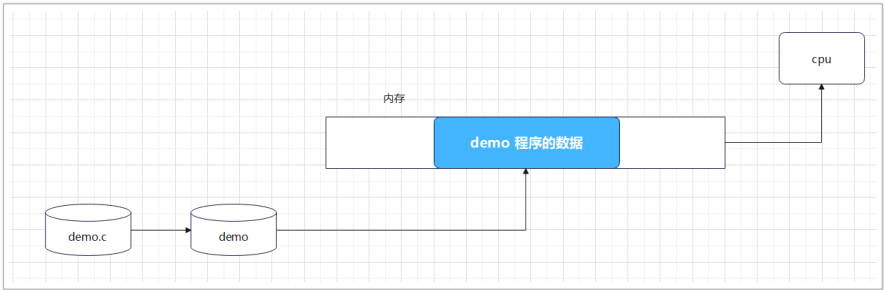
- 进程描述是一个程序执行过程，当程序执行后，执行过程开始，则进程产生, 执行过程结束，则进程也就结束
- 下面是 windows 系统中，所有的进程，即正在执行的程序，一旦结束进程，就意味着进程结束



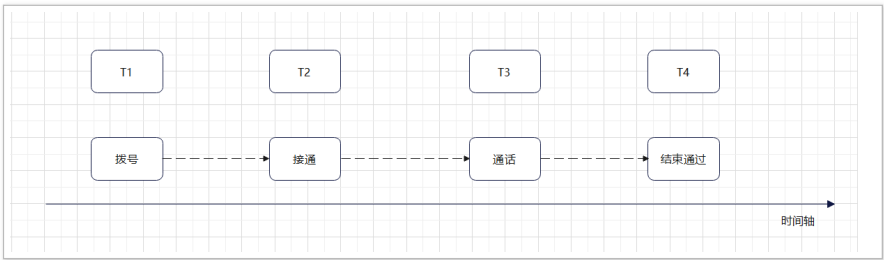
- 进程的特点如下:
 - 进程是一个独立的可调度的活动, 由操作系统进行统一调度, 相应的任务会被调度到cpu 中 j 进行执行



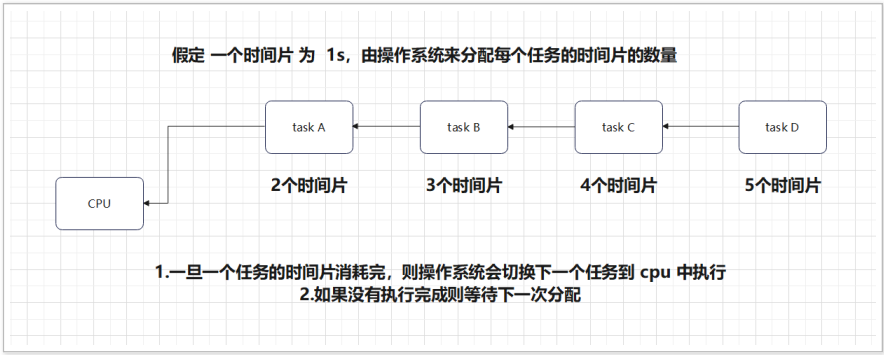
- 进程一旦产生，则需要分配相关资源，同时进程是资源分配的最小单位



- 进程和程序的区别
 - 程序是静态的，它是一些保存在磁盘上的指令的有序集合，没有任何执行的概念
 - 进程是一个动态的概念，它是程序执行的过程，包括了动态创建、调度和消亡的整个过程



- 并行执行：表示多个任务能够同时执行，依赖于 物理的支持，比如 cpu 是 4 核心，则可以同时执行 4 个任务
- 并发执行：在同一时间段有多个任务在同时执行，由操作系统调度算法来实现，比较典型的就 是时间片轮转



- 在 Linux 系统中管理进程使用
树型
管理方式
 - 每个进程都需要与其他某一个进程建立 父子关系, 对应的进程则叫做 父进程
- Linux 系统会为每个进程分配 id , 这个 id 作为当前进程的唯一标识, 当进程结束, 则会回收
- 进程的 id 与 父进程的 id 分别通过 getpid() 与 getppid() 来获取
- getpid 函数

函数头文件

```
#include <sys/types.h>
```

```
#include <unistd.h>
```

函数原型

```
pid_t getpid(void);
```

```
pid_t getppid(void);
```

函数功能

获取当前进程 id 与 父进程 id

函数返回值

成功 : getpid() 返回当前进程 id ,getppid() 返回父进程 id

pid_t 的类型 实际为 int , 在系统中采用 typedef 的形式

示例 : 获取当前进程 pid 与 ppid

```
#include <stdio.h>

#include <sys/types.h>

#include <unistd.h>

int main(void)
{
    printf(" pid : %d ppid : %d\n",getpid(),getppid());

    return 0;
}
```

~

练习:

1. 理解进程的概念与树型管理结构
2. 使用 `getpid()` 与 `getppid()` 获取当前进程

全文完

本文由 简悦 SimpRead 优化，用以提升阅读体验

使用了 全新的简悦词法分析引擎 beta，点击查看详细说明

