

CITS2002 Project 1 情报

Project 简介

本次编程项目的主要目的是通过读取文件 (tracefile) 中的硬件信息和任务详情，模拟一个搭配多个 I/O 设备的单核心 CPU 计算终端，并加以合理的调度分配。使用 C 语言写出模拟程序读取文件并计算出最优运行所需要的单次执行时间。

Requirement 需求

本项目主要考察的是学生对 C 语言的理解以及实际编程能力，套以具体的待解决问题使得本程序同操作系统进程分配等课程内容形成联系。在开始编程之前，请务必熟悉第八课的内容，了解 CPU 的工作模式和操作系统如何调度进程来充分利用 CPU。建议可以先在纸张上对简单的进程执行进行模拟，待对流程熟悉后使用具体的代码实现本程序。

Terms 术语

I/O Input and Output 输入及输出设备 如硬盘 外置存储 键盘 显示器 鼠标 打印机

Trace File 跟踪文件 记录运行状态的文件，包含已运行过的进程以及时间等信息。

Job Mix 进程运行的必备条件以及运行过程中可能产生的 I/O 请求的记录。

Time Quantum 单次执行时间 分配予各进程的时间，一旦时长结束，无论当前进程是否已经完成，都将其中断并放入等待队列。

Data Bus 数据总线 CPU 和其他 I/O 设备进行数据交流的途径

Model 模型

本项目模拟的是一个简单的运行在单核处理器上的操作系统，处理器的频率是 1GHz，但本项数据似乎并没有实际作用。所有运行的进程共有五个状态，除进程创建以及结束的 New 和 Exit 之外，更常见的是 Ready 排队等待运行，Blocked 等待 I/O 和 Running 正在运行的状态。

当一个进程开始运行，即从 New 转变为 Running 状态，有三种情况会导致状态切换。

1. 完成运行，准备退出，进入 Exit 状态
2. 运行时长超出单次执行时间，直接转为 Ready 状态开始排队，等待再次执行。
3. 发起 I/O 请求，转为 Blocked 状态，等待 I/O 设备完成处理后进入 Ready 队列。

Input 输入

程序读取的即为 Trace File 跟踪文件，对本文件的理解是编写程序代码的基础。

示例文件如下所示：

```
device      hd          80000000 bytes/sec
device      wifi        6750000 bytes/sec
device      screen      200000 bytes/sec
reboot
process 1 200 {
  i/o       100         hd          1600
  i/o       110         usb2        1600
  i/o       180         hd          1000
  i/o       190         usb2        1000
  exit      400
}
process 2 480 {
  i/o       8000        screen      40
  exit      8005
}
```

文件的第一部分是 I/O 设备的总结，名称以及传输速率。因为数据总线只有一个，所以传输速度最快的设备使用数据总线的优先级最先。速率是以比特每秒来表示的，与系统中所用的毫秒不同，务必注意。在完成设备信息的读取后，末尾的 Reboot 指令将会执行，系统时间归零并开始运行进程。每个进程的执行流程以一个区块表示，进程 ID 后的时间是开始执行进程的时间。区块内的时间是在进程执行一定时间后进行的 I/O 请求，请求设备以及请求数据大小。进程将会持续执行直到 EXIT 指令。

示例文件中展示的两个进程的执行流程如下：

进程 1 在系统启动 200 毫秒时开始执行，在执行 100 毫秒后发起对 I/O 设备 hd 总计 1600 比特的传输请求，待完成后另计 10 毫秒的执行（总执行 110 毫秒）发起 usb2 总计 1600 比特的传输请求。同理直到第 190 毫秒运行，还有另外两个 I/O 请求。完成之后，再执行 210 毫秒并在第 400 毫秒时结束运行。

进程 2 在系统启动 480 毫秒时开始执行，在执行 8000 毫秒后发起对 I/O 设备 screen 总计 40 比特的传输请求，完成之后再执行 5 毫秒就结束运行。

Output 输出

本程序的一个目的是衡量所写的拟操作系统的调度效率，针对不同的跟踪文件，需要记下从第一个进程开始运行直到所有的进程运行结束的总执行时长，即 total process completion time。本程序的另一个目的是对提供的追踪文件进行多次单次执行时间不同的测试，来寻求最短的总执行时长。因此，输出的格式为 best 最佳单次执行时间 最佳总执行时长。

Procedure 流程

正如示范文件中，在进程 2 于启动后 480 毫秒时开始运行，与此同时进程 1 尚未运行结束。CPU 和 I/O 设备就需要同时被不同的进程利用，本程序就需要完成如同操作系统的职责对进程进行调度。本程序的任务是以不同大小的单次执行时间（提示：可以使用 for loop 循环）重复不停的执行整个流程，最终得到最短的总执行时长。

Arguments 参数

程序可以接受 2/4 个参数，第一个参数是跟踪文件的位置，当提供两个参数时，程序只会按照给定的单次执行时间（即参数 2）进行模拟并给出总执行时长，完成情况是四个参数，除去跟踪文件分别是单次执行时间范围起始，范围终止以及每次增加的时长。重复的模拟以下述 For Loop 表示。

```
for(int time_quantum=TQ0 ; time_quantum<=TQfinal ; time_quantum += TQinc)
```

在程序运行的起始，首先检查命令行参数，计入正确的单次执行时间循环范围。然后进行追踪文件的读写，将 I/O 设备以及进程与其流程记入内存中的数据结构。紧接着开始执行不同单次执行时间的循环，完成后输出最佳解。

Sections 任务分节

Design 设计

1. 决定如何储存 I/O 设备信息以及不同的进程队列
2. 决定如何储存所有的进程及其流程
3. 讨论如何完成模拟进程

Coding 编程

1. 完成存储设备和进程信息的数据结构
2. 完善 parse_tracefile() 即写入设备和进程信息
3. 完成 simulate_job_mix() 即模拟进程

Common Mistakes 易错点

TBA

Design 程序设计

Data Structure 数据结构

I/O 设备信息存放于 devSpeed 和 devNames 两个数组中，每一个设备依照速度快慢被分配唯一 ID。可以通过 devSpeed[ID] 或 devNames[ID] 访问其属性。在读取文件并加入设备时，使用其速度与已知设备进行比较，找到设备列表中适合的位置。