

Q1. What is a process? What is a program? And what is the difference?

说明进程与程序的区别：

process: 进程，是程序的一次执行（程序，数据以及 PCB），是系统资源（CPU、内外存、I/O 等）分配和调度的最小单元

program: 程序，为执行某一任务的静态指令集合，存在外存中的文件里

difference:

1. 进程是动态的，是一次执行过程，而程序是静态的
2. 进程是暂时的，有生命周期，程序是永存的
3. 进程可以并发，且进程是资源分配的最小单位
4. 一个进程可以对应一个或多个程序，一个程序也可以对应一个或多个进程

Q2. What is job? And what is the difference between process and job?

说明进程与作业的区别：

job: 作业，是提交给操作系统的工作集合（一组程序）。作业的完成分为提交、收容、执行和完成四步。

process: 见上一题

difference:

1. 作业是任务实体，进程是执行实体
2. 作业队列（job queue）在外存中，进程总有部分存在于内存中
3. 作业包含至少一个或者更多进程，进程不能对应多个作业

Q3. What are the differences between job scheduling and process scheduling?

说明作业调度与进程调度的区别：

job scheduling: 作业调度（高级调度），按一定的原则从外存中的 job queue 中选取作业加载到内存中的 ready queue 上，为其生成相应的进程并提供资源。平衡操作多程序执行期间各种资源的占用（CPU、I/O 资源）。

process scheduling: 进程调度（低级调度），按一定的算法把 ready queue 中的进程交给 CPU 执行

difference:

1. 作业调度速度比进程调度慢
2. 作业调度控制 degree of multiprogramming, 进程调度减少 multiprogramming
3. 作业调度将作业从外存中加载到内存中并生成对应进程, 进程调度可以将进程重新引入内存然后让 CPU 继续执行这个进程

Q4. How many statuses are in a job(in the code for this report—scheduler)? And what are they?

本次实验作业有几种状态(在本次报告的代码中-scheduler)? 请列举, 并分别解释几种状态

共有三种状态: READY, RUNNING, DONE

READY: 作业准备就绪, 可以被执行

RUNNING: 作业正在被执行

DONE: 作业完成, 可以被移除

Q5. What programs for job control are used in this experiment? And their function?

本实验作业控制命令处理程序包括哪些? 它们分别实现什么功能?

控制命令处理程序共有三种: ENQ, DEQ, STAT

ENQ: 向调度程序发送入队请求, 使作业进入就绪队列, 并初始化其各类属性, 为其创建进程、设置状态为 READY

DEQ: 向调度程序发送出队请求, 使作业离开就绪队列, 并释放对应内存空间

STAT: 在标准输出中打印所有处于 READY, RUNNING 作业的状态, 包括 job id, job pid, job owner, job run time, job wait time, job create time 和 job state (修改后, 还有 jobname, current priority, default priority)

Q6. What is used for inter-process communication in this experiment? And its function?

本实验采用什么进行进程之间的通信? 它相当于什么作用?

使用有名管道 (FIFO) 进行通信。

作用: 创建的有名管道的路径, 管道以文件的形式存在于文件系统中, 不会随程序的终止而

消失，为有文件路径权限的进程提供通信。FIFO 可以使得毫不相关的进程互相通信，常用于多进程通信。在本次作业中，控制命令处理程序把命令写入 FIFO 文件，调度程序 scheduler 从 FIFO 文件中读出命令。

Q7. Run the job scheduler program, And analyze the execution of the submitted job:

运行作业调度程序，分析提交作业后的作业的执行过程:

作业提交后，为其生成对应的进程、优先级 以及其他属性，将其放入就绪队列中等待执行。每个作业默认 优先级由用户设定，随后每等待 100ms 其优先级加一（不超过 3），通过调度 完成一次执行后，优先级重置为默认优先级。通过调度程序控制，使得每个作业一次执行的时间不超过一个时间片，随后在就绪队列中选择优先级最高（优先级相同时选择等待时间更长的作业）的作业进行下一次的执行。若原作业此时仍未完成执行，则其回到就绪队列中，等待下一次的调度执行。若原作业此时仍未完成执行，则其回到就绪队列中，等待下一次的调度执行。接收到出队命令后作业会（若正在执行则先停止执行）离开就绪队列，并清空相关数据结构。

当接受到显示状态的命令时，正在执行的作业以及就绪队列中的队列信息会被输出到标准输出。

Q8. Understand the process of job scheduling —— Submit a new job (Execution results and corresponding code):

Schedueler 作业调度的过程理解——提交新作业（分析执行结果及代码表现）:

向调度程序发送新作业的入队请求，为其分配 job id, 创建对应进程，初始化该 job 的各类属性，设置其状态为 READY，并将其放入就绪队列，等待调度程序 scheduler 调度其执行。作业完成后该 job 不再进入就绪队列，并释放占用的资源。

Q9. Understand the process of job scheduling —— End of job execution (Execution results and corresponding code):

Schedueler 作业调度的过程理解——作业正常执行结束（分析执行结果及代码表现）:

作业正常执行结束后，其 state 为 DONE，在标准输出中输出对应提示语。同时，该作业离开就绪队列并释放占用的资源。

Q10. Understand the process of job scheduling —— job scheduling due to Priority(Execution results and corresponding code):

Schedueler 作业调度的过程理解——因为优先级和进行作业调度（分析执行结果及代码表现）:

当处于 RUNNING 状态的作业时间片耗尽或者执行完成而正常终止时，进行调度，选择另一

个作业进行执行。若原作业此时未完成执行，则回到就绪队列中，等待下一次的执行。

Q11. 因为时间片进行作业调度