E08：UNIX 进程控制二

参考答案与说明

一、简答题：

1. 当以下标志出现时，UNIX 会产生什么动作？

【参考答案】

（1）RunRun==1，为强制调度标志，表示系统里现在出现了比现运行进程更适合占用处理机的进程。在下一次有进程做例行调度时（一次先前态为用户态的中断返回，检查 RunRun），将会调用 Swtch，实施一次进程调度。

（2）RunIn==1，表示盘交换区上有进程需要进入内存，但是内存没有足够的空间可供调入，且寻找不到可供换出的进程图像。此时，0#进程因为 RunIn 进入睡眠状态。未来，如果内存出现可供换出的进程（有进程进入低睡状态或 1 秒计时到），0#进程被唤醒，重新寻找可供换出的进程图像。

（3）RunOut==1，表示盘交换区上没有就绪状态的进程需要进入内存。此时，0#进程因为 RunOut 进入睡眠状态。未来，如果有一个盘交换区上睡眠的进程被唤醒，将同时唤醒 0#进程，醒来的 0#进程上台后，将该进程的图像换入内存。

二、应用题

1.

【参考答案】

（1）T0 时刻，现运行进程 pa 执行 read 系统调用：进程 pa 由于执行磁盘 I/O，调用内核函数 Sleep，进入高睡状态，放弃处理器。Sleep 中调用 Swtch，选中内存中的就绪进程 pb，使其占用处理器继续执行。

（2）T1 时刻，pa 启动的 I/O 操作完成：pa 启动的 I/O 操作完成，正在 CPU 上运行的进程 pb 响应中断请求。在磁盘中断处理中，唤醒进程 pa。进程 pa 进入就绪状态，等待下一次被调度到后上台执行。

（3）T2 时刻，pa 正在执行时，pc 等待的 I/O 操作完成，因为此时盘交换区只有一个低睡进程，所以 0#进程一定因为 RunOut 在睡眠：

- pa 响应中断，唤醒 pc 进程，SetRun 函数中唤醒因为 RouOut 睡眠的 0#进程；
- pa 返回用户态时例行调度，0#进程上台执行 Sched，找到进程 pc，为其申请内存；
- 如果申请成功，将 pc 进程的图象调入，释放盘交换区空间，修改 p_addr，设置 SLOAD 标志；
- 如果申请不成功，0#进程选择可以换出的进程；
- 如果找到可以换出的进程（pa 或 pb），为该进程申请盘交换空间，调出，释放内存，将 pc 进程的图象调入（如果换出一个不够，有可能换出 2 个进程）；
- 如果找不到可以换出的进程，0#进程因为 RunIn 睡眠，未来有进程进入低睡或 1 秒计时到，将 0#进程唤醒，再次尝试先换出，再换入；
- 何时进程 pc 进入内存，将等待 Swtch 的调度。

2.

【参考答案】

（1）进程 p1 由于执行 I/O，调用内核函数 Sleep，进入高睡。Sleep 中调用 Swtch，选中内存中的就绪进程 p2，使其占用处理器继续执行。

序号	占用空间	状态	位置	年龄（p_time）

0#	-	高睡（RunOut）	SLOAD	-
p1	40K	高睡	SLOAD	2
p2	30K	执行	SLOAD	3
p3	30K	低睡	~SLOAD	3

（2）由于 p3 等待的 I/O 操作完成，现运行进程 p2 执行中断处理程序，唤醒 p3，因 p3 图象在盘交换区，且 RunOut 被设置，则同时唤醒 0#。由于唤醒了高优先级进程 0#，RunRun 被设置。p2 中断返回的例行调度中，0#上台。各个进程状态的变化如下表所示：

序号	占用空间	状态	位置	年龄（p_time）
0#	-	执行	SLOAD	-
p1	40K	高睡	SLOAD	3
p2	30K	就绪	SLOAD	4
p3	30K	就绪	~SLOAD	4

0#进程上台后继续执行 Sched 程序：

- 找到盘交换区的 p3 进程，由于内存没空，所以为 p3 内存分配不成功；
- 找可以换出的进程。由于内存没有低睡进程，且 p3 进程的年龄>3 秒，则找高睡和就绪状态的进程；找到进程 p1，因为 p1 的年龄>2 秒，所以将 p1 换出；
- 尝试为 p3 进程分配内存，成功，p3 进程换入；

此时，各个进程状态的变化如下表所示：

序号	占用空间	状态	位置	年龄（p_time）
0#	-	执行	SLOAD	-
p1	40K	高睡	~SLOAD	0
p2	30K	就绪	SLOAD	4
p3	30K	就绪	SLOAD	0

再无需进内存的进程，0#进程设置 RunOut 之后，执行 Sleep，进入高睡；Sleep 中调用 Swtch，由于 p2 是计算获得的优先数，而 p3 设置获得的优先数，则选中内存中的就绪进程 p3。各个进程状态的变化如下表所示：

序号	占用空间	状态	位置	年龄（p_time）
0#	-	高睡（RunOut）	SLOAD	-
p1	40K	高睡	~SLOAD	0
p2	30K	就绪	SLOAD	4
p3	30K	执行	SLOAD	0

3.

【参考答案】

（1）fork()之后，父进程执行语句为：

```
    {
            a=a+1;
            printf(" i= %d; a= %d\n", i, a);
    }
}
```

子进程执行语句为：

```
    {
```

```
            a=a+2;
            printf(“ i= %d; a= %d\n”, i, a);
    }
}
```

（2）程序的输出结果可能有下列 2 种情况：

i=505；a=1
i=0; a=2

或

i=0；a=2
i=505; a=1

4.

【参考答案】

程序的输出结果如下：

It is child process.
It is parent process.
The finished child process is 505.
The exit status is 1.