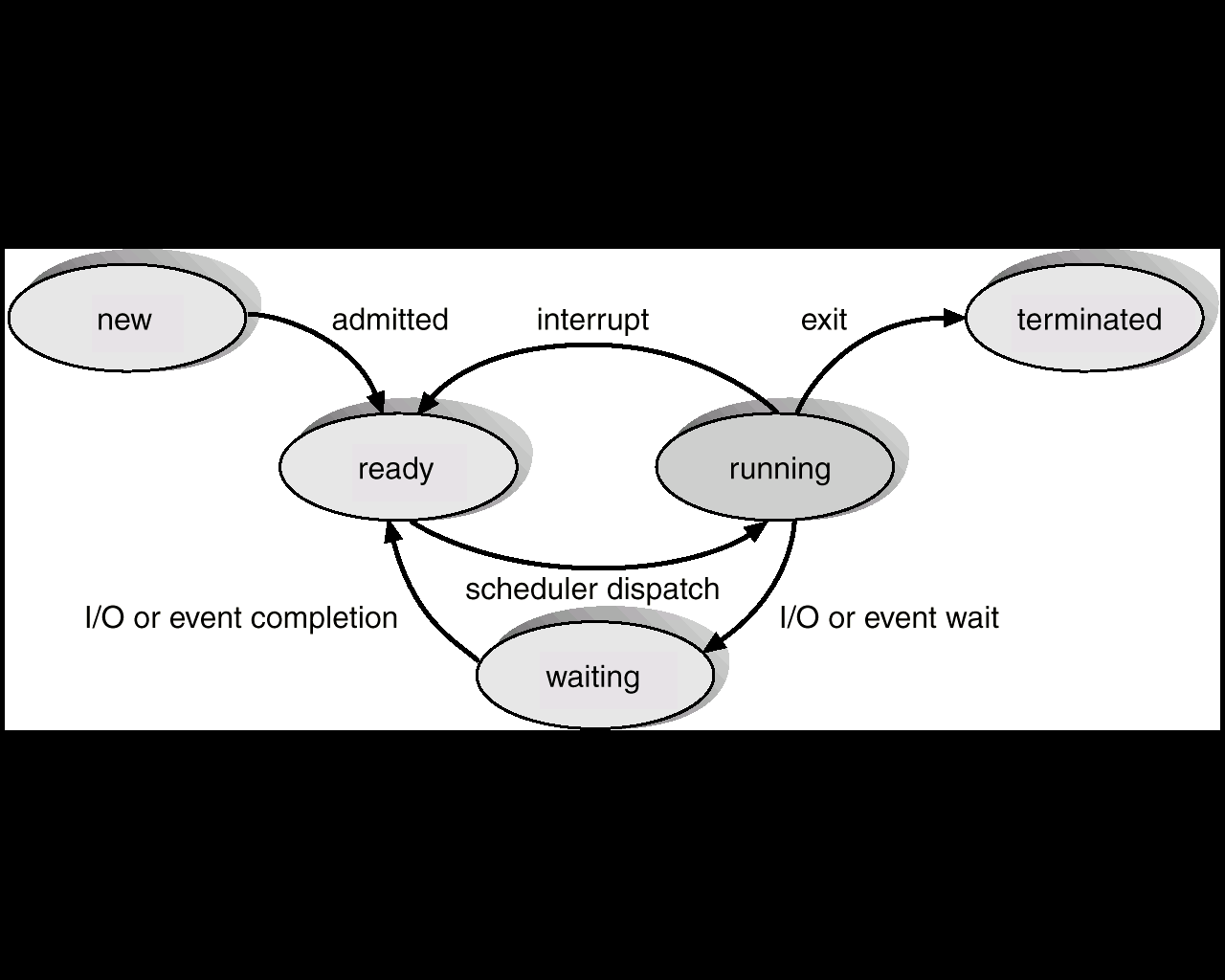
1. (20%) 简述进程的概念。画出进程的状态图，并就状态图中可能发生CPU调度决策的四种情况进行说明。结合进程控制块PCB对内核在两个进程间进行上下文切换过程进行描述。

**参考答案：进程，执行中的程序；执行一个程序所需要的资源集的系统抽象；一个程序的运行实例；**

**进程状态图：**

****

**CPU调度决策可以如下四种情况下发生：**

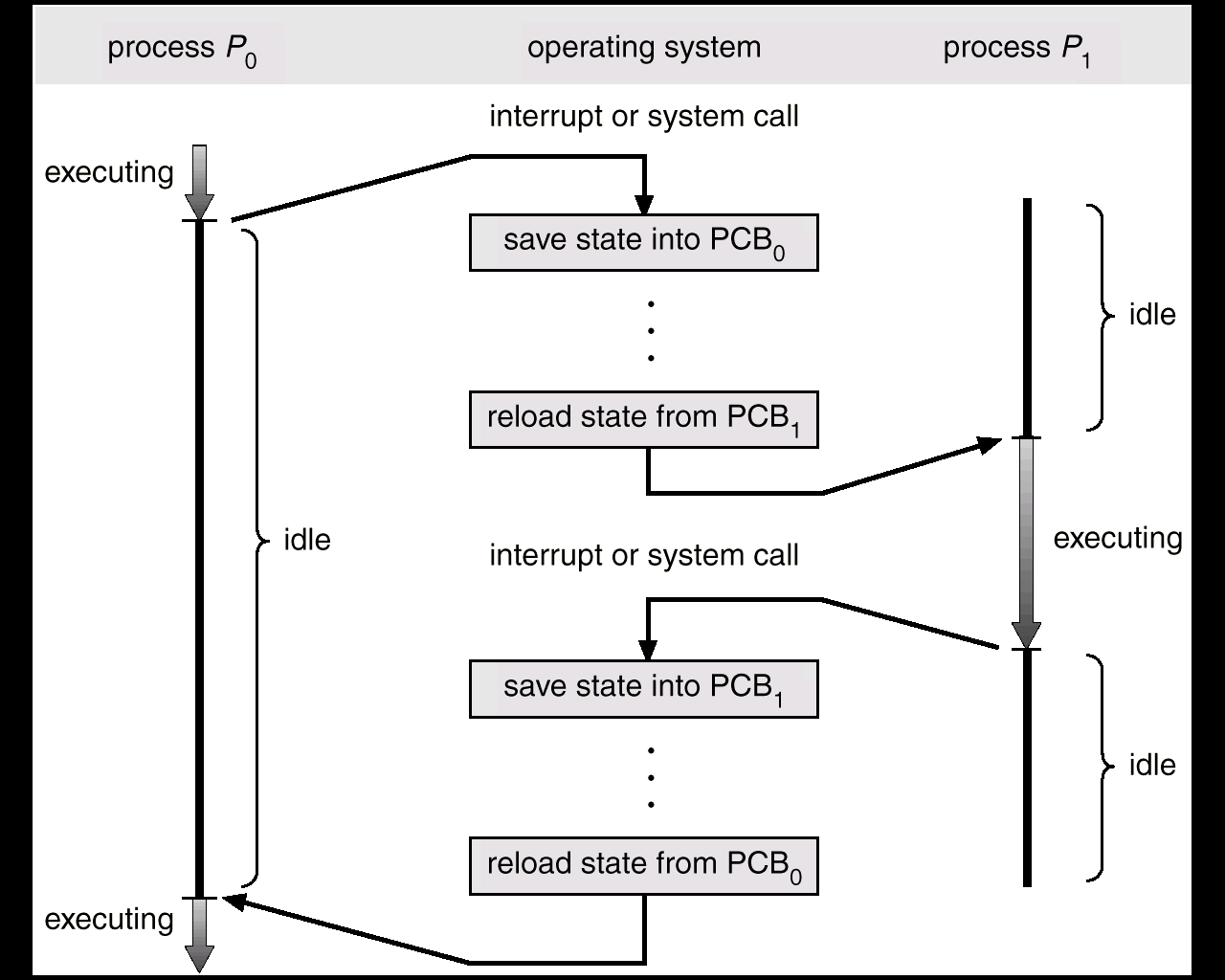
**当一个进程从运行状态切换到等待状态**

**当一个进程从运行状态切换到就绪状态**

**当一个进程从等待状态切换到就绪状态**

**当一个进程终止时。**

**进程切换：**



2. (10%) 简述系统调用（System call）、系统程序（System program）的用途。

**参考答案：系统调用，允许用户进程请求操作系统提供的服务。由OS内核实现并提供访问接口，由用户程序调用，程序员使用。**

**系统程序，给用户提供基本的功能，这样用户在解决公共问题的时候不用写自己的程序。提供程序开发和执行的环境，普通操作用户使用。**

3. (15%) 线程通常可分为用户级线程和内核级线程，请回答两种线程之间的区别，并就两者之间的关系，对多线程模型的类型及其优缺点进行简要说明。

**参考答案：用户线程不受内核支持，无需内核管理，使用用户级的线程库完成线程的管理，对用户线程的支持通常处于内核之上，通过一个用户级线程库（thread library）实现。线程库提供了对线程的创建、调度和管理的支持，这无需来自内核的支持。因为内核并不知道用户级线程的存在，所有的线程创建和调度工作都在用户空间完成，而且整个过程不受内核的干涉。**

**内核线程由操作系统直接支持：内核在内核空间内实现了线程的创建、调度和管理。**

**常见的多线程模型包括以下三种：**

**多对一， 优点线程管理由线程库在用户空间进行，效率较高，缺点一旦某个线程执行了阻塞，则整个进程也会被阻塞**

**一对一， 允许线程并发的执行并且运行在多处理器系统上，不会导致单一线程阻塞进程的情况，缺点用户线程和内核线程一一对应，开销较大，限制了系统线程的数量。**

**多对多， 多路复用了许多用户级线程到同样数量或更小数量的内核线程上，允许OS创建足够数量的内核线程，且在某个线程发生阻塞时，可调度其他线程执行。**

4. (15%) 对于一个多任务的应用软件，分析比较采用多进程实现和多线程实现的优劣，并举例说明。

**仅供参考：从资源或系统开销、维护、进程阻塞、崩溃、安全等情况分析。**

**以web浏览器为例（360浏览器从多线程到多进程实现）。**

5. (10%) 如下所示程序使用Pthread API实现，请将空白部分补充完整，并说明LINE C和LINE P将会输出的结果。

**#include <pthread.h>**

**#include <stdio.h>**

**int value = 0;**

**void \*runner(void \*param); /\* the thread \*/**

**int main{int argc, char \*argv[])**

**{**

**int pid;**

**pthread\_t tid;**

**pthread\_attr\_t attr;**

**pid = fork () ;**

**if (pid == 0) {/\* child process \*/**

**pthread\_attr\_init ( 1 ) ;**

**2 (&tid, &attr , runner, NULL) ;**

**3 (tid, NULL) ;**

**printf("CHILD: value = %d",value); /\* LINE C \*/**

**}**

**else if (pid > 0) {/\* parent process \*/**

**wait(NULL);**

**printf("PARENT: value = %d",value); /\* LINE P \*/**

**}**

**}**

**void 4 (void \*param){**

**value = 5;**

**pthread\_exit (0) ;**

**}**

**参考答案：**

**1、&attr**

**2、pthread\_create**

**3、pthread\_join**

**4、\*runner**

**LINE C、CHILD: value =0**

**LINE P、PARENT: value =5**

6. (20%) 有5个待运行的作业A、B、C、D、E，它们的运行时间分别为10、6、2、4和8分钟，其提交时间分别为第0、2、3、5、7分钟，其优先级别分别为3、2、5、1、4，其中5级为最高优先级。对于下列进程调度算法，试计算其平均周转时间。

1）、先来先服务；

2）、非抢占式优先级调度；

3）、抢占式优先级调度。

**参考答案：**

**1）**

E

D

C

B

A

**0 10 16 18 22 30**

**平均周转时间15.8分，平均等待时间9.8分。**

**2）**

E

B

D

C

A

**0 10 12 20 26 30**

**平均周转时间16.2分，平均等待时间10.2分。**

**3）**

A

E

A

C

A

B

D

**0 3 5 7 15 20 26 30**

**平均周转时间15.8分，平均等待时间9.8分。**

7. (10%) 阅读下列代码，回答A、B、C、D、E处代码的作用。

ORG 0x7c00

entry:

MOV AX,0

MOV SS,AX

MOV SP,0x7c00

MOV DS,AX

MOV ES,AX

MOV AX,0x0012 ; A设置显示模式

INT 0x10 ; B调用显示的BIOS

MOV SI,msg

putloop:

MOV AL,[SI]

ADD SI,1

CMP AL,0

JE fin

MOV AH,0x0e

MOV BL,0x02 ;C 指定字符颜色（IRGB）

INT 0x10

JMP putloop

fin:

HLT

JMP fin

msg:

DB 0x0a

DB "Hello, Demo OS boot sector!"

DB 0x0a

DB 0

RESB 0x7dfe-$ ;D 填写0x00直到0x7dfe

DB 0x55, 0xaa ;E 设置可引导扇区标志