**第二章**

1. \_\_\_\_优先权是在创建进程时确定的，确定之后在整个进程运行期间不再改变。

动态

先来先服务

静态

短作业

2.

2. 下列进程状态变化中，\_\_\_\_\_\_变化是不可能发生的。

运行—>等待

等待—>运行

运行—>就绪

等待—>就绪

3.

3. 当\_\_\_\_\_时，进程从运行状态变为就绪状态。

等待某一事件

进程被调度程序选中

时间片到

等待的事件发生

4.

4. 进程管理中，当\_\_\_\_\_，进程从阻塞态变成就绪态。

等待的事件发生

等待一个事件

时间片用完

进程被进程调度程序选中

5.

5. 下面对进程的描述中，错误的是\_\_\_\_。

进程是动态的概念

进程是指令的集合

进程是有生命周期的

进程执行需要处理机

6.

6. 下面所述步骤中，\_\_\_\_\_不是创建进程所必需的。

建立一个进程控制块

由调度程序为进程分配CPU

将进程控制块链入就绪队列

为进程分配内存

7.

7. 多道程序环境下，操作系统分配资源以\_\_\_\_为基本单位。

指令

作业

进程

程序

8.

8. 下述哪一个选项体现了原语的主要特点\_\_\_\_\_\_\_。

并发性

异步性

共享性

不可分割性

9.

9. 关于内核级线程，以下描述不正确的是\_\_\_\_\_\_。

可以将一个进程的多个线程分派到多个处理器，能够发挥多处理器并行工作的优势

控制权从一个线程传送到另一个线程时不需要用户态-内核态-用户态的模式切换

建立和维护线程的数据结构及保存每个线程的入口

内核可以将处理器调度直接分配给某个内核级线程

10.

10. 一个进程被唤醒意味着\_\_\_\_。

进程变为就绪状态

其PCB移至等待队列队首

该进程重新占有了CPU

它的优先权变为最大

11.

11. 在引入线程的操作系统中，资源分配的基本单位是\_\_\_\_。

程序

线程

作业

进程

12.

12. 在下述关于父进程和子进程的叙述中，正确的是\_\_\_\_\_。

父进程和子进程可以并发执行

父进程创建了子进程，因此父进程执行完了，子进程才能运行

撤销父进程时，应该同时撤销子进程

撤销子进程时，应该同时撤销父进程

13.

13. 对进程的管理和控制使用\_\_\_\_\_\_\_。

原语

指令

信号量

信箱通信

14.

14. 所谓“可重入”程序是指\_\_\_\_\_\_。

能够被多个进程共享的程序

不能够被多个程序同时调用的程序

在执行过程中其代码自身会发生变化的程序

无限循环程序

15.

15. 原语是\_\_\_\_\_\_。

操作系统的内核

不可中断的指令序列

运行在用户态下的过程

可中断的指令序列

16.

16. 在进程调度算法中，对短进程不利的是\_\_\_\_\_。

先来先服务算法

短进程优先调度算法

多级反馈队列调度算法

高响应比优先算法

17.

17. 一个可共享的程序在执行过程中是不能被修改的，这样的程序代码应该是\_\_\_\_\_。

可执行代码

可重入代码

封闭的代码

可再现代码

18.

18. 在进程管理中，当\_\_\_\_\_时，进程状态从运行态转换到就绪态。

等待的事件发生

时间片用完

等待某一事件发生

进程被调度程序选中

19.

19. Solaris的多线程的实现方式为\_\_\_\_\_\_。

纯用户级多线程

纯内核级线程

单线程结构进程

混合式

20.

20. 在UNIX系统中运行以下程序，最多可再产生出\_\_\_\_\_\_进程?

main( ){

fork( ); /\*←pc(程序计数器)，进程A

fork( );

fork( );

}

5

7

3

9