homework03

3.1

- **Short-term（CPU调度器）**：从内存中选择那些准备好执行的作业，并将CPU分配给它们。
- **Medium-term**：特别用于分时系统，作为一个中间调度级别。实施了一种交换方案，将部分运行的程序从内存中移除，并在稍后恢复它们，以便它们可以从中断的地方继续执行。
- **Long-term（作业调度器）**：决定哪些作业被带入内存进行处理。
- 它们之间的主要区别在于它们的执行频率。短期调度必须经常选择一个新进程。长期调度使用得少得多，因为它处理将作业放入系统，并且可能需要等待一个作业完成后才会接纳另一个作业。

3.2

通常情况下，操作系统必须保存当前运行进程的状态，并恢复下一个计划运行进程的状态。保存进程的状态通常包括所有CPU寄存器的值以及内存管理信息。上下文切换还必须执行许多特定于架构的操作，包括清空数据和指令缓存。
当发生上下文切换时，内核会将旧进程的关联状态保存在其 PCB 中，然后装入经调度要执行的新进程的已保存的关联状态。

3.4

1. 定义了一个全局变量 `value`，并初始化为 5
2. 在`main`函数中，调用`fork()`创建一个新的进程
3. 如果 `pid` 等于 0，说明当前是子进程，将 `value` 增加 15，此时子进程中的 `value` 变为 20
4. 如果 `pid` 大于 0，说明当前是父进程，父进程调用 `wait(NULL)` 等待子进程结束
5. 父进程在 `wait` 之后打印 `value` 的值，然后退出
   由于 `fork()` 创建子进程时，子进程会复制父进程的内存空间，包括全局变量 `value`。但是，子进程对 `value` 的修改不会影响到父进程的 `value`。因此，在父进程中，`value` 的值仍然是初始值 5。
   所以输出是：

```c
PARENT: value = 5
```