

```

// 进程状态枚举
enum ProcessStatus {
    READY,
    RUNNING,
    WAITING,
    ENDING
}; // 0, 1, 2, 3

// 进程控制块 PCB 结构体
struct PCB {
    int pid; // 进程 ID
    int aTime; // 到达时间
    int bTime; // 预计完成时间
    ProcessStatus status; // 进程状态
    int priority; // 进程优先级
};

struct TotalChain {
    // 用链表结构
    PCB* runningPcb; // 运行队列，只有一个
    std::list<PCB*> readyList; // 就绪队列
    std::list<PCB*> waitList; // 等待队列
    std::list<PCB*> freeList; // 空闲队列
}

// PCB 管理类
class PCBManager {
private:
    // 空闲 总链 list 就绪队列 等待队列
    // 空闲队列和总链加起来就是全部的PCB的空间
    TotalChain tchain; // 总链
public:
    // 构造函数
    PCBManager(); // 默认初始值--空闲队列满，其他队列空

    // 创建、撤销、时间片到、挂起进程和激活进程等基本进程原语---函数
    // 创建进程：分配一个新的PCB，并初始化其状态。就是从空闲队列上取，放到总链上--其实我还是不太
    // 知道要干啥
    void createProcess(int pid, int priority);
    // 撤销进程：清理PCB并释放相关资源，放回空闲队列上。
    void undoProcess(int pid);
    // 时间片到：将当前运行的进程移回就绪队列，并从就绪队列中选择下一个进程运行（FIFO）。--
    void timeOut(int pid);
    // FIFO：从就绪队列中选择下一个进程运行
    void FIFO();
    // 运行：就是表示正在运行，展示可视化菜单，是挂起 结束 还是时间片到
    void running(int pid);
    // 挂起进程：将进程从就绪队列移动到等待队列。 ???这是我有疑问的地方，就是怎么表示运行的状态，
    // 怎么触发挂起
    void suspendProcess(int pid);
    // 激活进程：将进程从等待队列移回就绪队列。
    void activateProcess(int pid);
    // 打印快照（当前运行的 就绪队列 等待队列 pid就可以了）
    void print();
    //

```

```
};  
int main(){  
    //实现菜单逻辑  
  
    return 0;  
}
```