

## 1. 进程是什么？

1. 进程是一个具有一定独立功能的程序关于某个数据集合的一次运行活动
2. 进程是操作系统进行资源分配和调度 的一个独立单位
3. 包括 OS 管理控制运行程序的数据结构、程序状态字信息，运行程序的内存代码、内存数据、通用寄存器信息

## 2. 进程表是什么？

```
PUBLIC PROCESS proc_table[NR_TASKS];
```

1. OS 用于记录和刻画进程状态及环境信息的数据结构，包含标识信息、现场信息、控制信息
2. 借助进程表，OS 可以全面管理进程的物理实体，刻画进程的执行现状，控制进程的执行

## 3. 进程栈是什么？

1. 进程运行时自身的堆栈，包括用户栈和核心栈
2. 进程在用户态下运行使用用户栈
3. 进程在内核态下运行使用内核栈

## 4. 当寄存器的值已经被保存到进程表内， esp 应该指向何处来避免破坏进程表的值？

esp 指向专门的内核栈区域

## 5. tty 是什么？

计算机的终端(terminal)设备

## 6. 不同的 tty 为什么输出不同的画面在同一个显示器上？

不同 TTY 各有一个 CONSOLE，各个 CONSOLE 公用同一块显存。

虽然不同的 TTY 对应的输入设备是同一个键盘，但输出却好比是在不同的显示器上，因为不同的 TTY 对应的屏幕画面可能是迥然不同的。实际上，我们当然 是在使用同一个显示器，画面的不同只不过是因为显示了显存的不同位置罢了

## 7. 解释 tty 任务执行的过程?

1. TTY 初始化, 开始循环
2. 对于每一个 TTY 先执行 `tty_do_read()`, 它调用 `keyboard_read()` 并将读入的字符交给 `in_progress()` 处理
3. 如果是要输出的字符, 会被 `in_progress()` 放入当前接受处理的 TTY 的缓冲区
4. 再执行 `tty_do_write()`, 如果缓冲区有数据, 就被送入 `output_char()` 显示出来

## 8. tty 结构体中大概包括哪些内容?

1. 输入缓冲区及其头尾指针和填充的大小
2. 指向 console 的指针

## 9. console 结构体中大概包括哪些内容?

1. 当前显示到的位置、光标位置
2. 当前控制台对应的显存位置和占用的大小

## 10. 什么是时间片?

时间片是分时操作系统分配给每个正在运行的进程微观上的一段 CPU 时间

## 11. 结合实验代码解释什么是内核函数? 什么是系统调用?

系统调用是用户进程进入内核的接口层, 它是由内核函数实现的, 进入内核后, 不同的系统调用会找到相应的内核函数, 这些内核函数被称为系统调用的“服务例程”。

内核函数与普通函数形式上没有什么区别, 只不过前者在内核实现, 因此要满足一些内核编程的要求。

用户空间的程序无法直接执行内核代码, 因为内核驻留在受保护的地址空间上, 不允许用户进程在内核地址空间上读写。

例如, `sys_sleep` 为内核函数 (`proc.c`), `syscall.asm` 中的 `mySleep` 即为系统调用