异常处理程序的结构

（1）**准备阶段**：在内核栈保存通用寄存器内容。

（2）**处理阶段**：保存硬件出错码和异常类型号，然后向当前进程发送一个信号。

异常处理程序的结构

（3）**恢复阶段**：恢复保存在内核栈中的各个寄存器的内容，切换到用户态并返回到当前进程的断点处继续执行。

信号处理函数

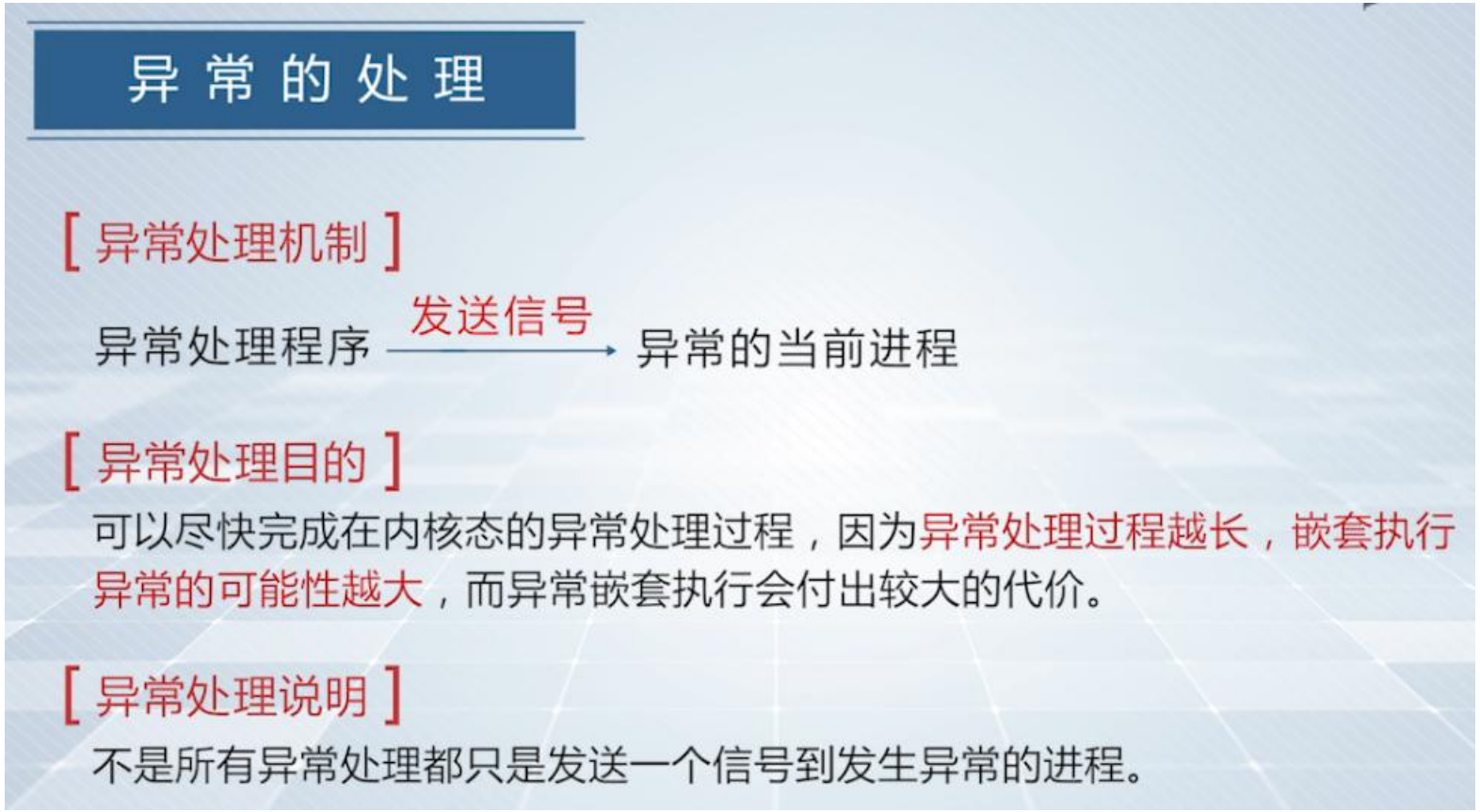
» Linux对于每个信号名都有一个默认的处理函数；

» 也可以自定义一个处理函数，将其与某个信号绑定。

[说 明]

异 常 的 处 理

[异常处理举例]



例如对于上一页的例子中，对于不可恢复的故障如越级访问，页故障处理程序才发送SIGSEGV信号给进程；如果只是缺页这种异常，页故障处理程序会把所缺失的页面从磁盘装入主存，返回到发生缺页故障的指令处继续执行

setjmp和longjmp函数

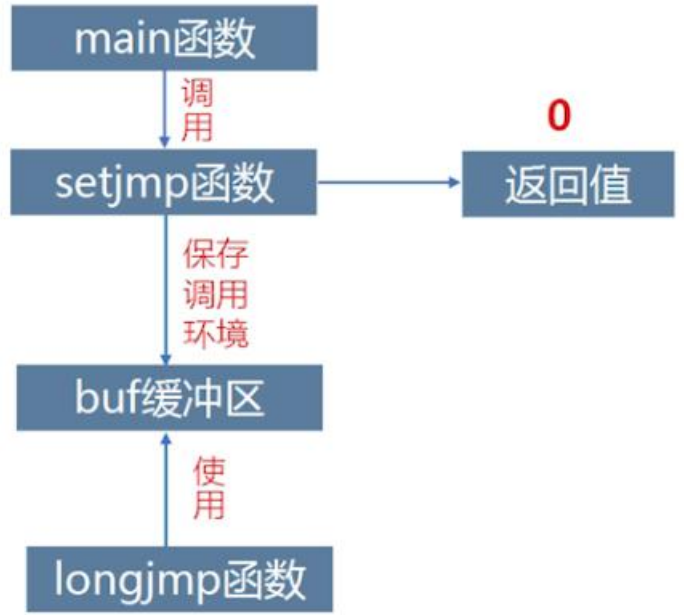
setjmp函数第一次被调用时返回一次，每一次调用longjmp时又会返回一次。

longjmp函数被调用时，永远不会返回。

setjmp(env) 函数： 将程序中的上下文存储在**env**中，包括程序计数器、栈帧、通用寄存器等。

longjmp(env, status)函数： **env**指代 **setjmp**所保存的函数执行状态的变量，**status**用于让**setjmp**函数返回的值。

说明：setjmp调用一次，返回多次；
longjmp调用一次，从不返回。



非本地跳转(nolocal jump)

c语言提供的一种用户级异常控制流的形式，将控制权直接从一个函数转移到当前正在执行的另一个函数，而无需执行正常的调用和返回序列。

由setjmp和longjmp函数提供。

非局部跳转的应用

允许从多层嵌套的函数调用中立即返回，通常是由于检测到某些错误条件。如果在嵌套函数调用的深处检测到错误条件，则可以使用非本地跳转直接返回到通用的本地化错误处理程序，而不是展开调用堆栈。