

# 事件驱动编程

## 实验目的：

掌握 Unix/Linux 环境下事件驱动编程的一般方法，能利用 curses 库、间隔计时器（interval timer）、信号处理编写一个功能完整的视频游戏程序，并能在 Unix/Linux 环境下正确地运行。

## 实验内容与要求：

使用 curses 函数进行屏幕管理、使用计时器进行时钟管理、利用信号机制进行中断管理、编写一个功能完整的视频游戏（自行设计）。

### 1.使用 curses 函数进行屏幕管理

编译和运行 hello1.c、hello2.c。记录实验结果。

如: \$ gcc hello1.c -l curses -o hello1  
\$./hello1

### 2. 时钟编程

(1)时钟编程: sleep

(a)分析源代码 hello3.c、hello4.c。

(b)编译运行 hello3.c 和 hello4.c。记录实验结果。

(2)时钟编程: alarm

(a)分析源代码 sleep1.c。

(b)编译运行 sleep1.c。记录实验结果。

(3)使用计时器进行时钟管理（间隔计时器(interval timer)）

(a)分析源代码 ticker\_demo.c。

(b)编译运行 ticker\_demo.c。记录实验结果。

### 3.利用信号机制进行中断管理

(1) 分析源代码 sigdemo.c。

(2) 编译并运行 sigdemo.c，测试多个信号。

程序运行时，输入以下几种情况(要输入得快)，记录输出结果。

(a)[Ctrl+c][Ctrl+c][Ctrl+c] [Ctrl+c]

(b)[Ctrl+c][Ctrl+\]

(c)hello[Ctrl+c][Enter]

(d) hello[Enter] [Ctrl+c]

(e)[Ctrl+\][Ctrl+\]hello[Ctrl+c]

### 4.使用计时器和信号：视频游戏

(1)分析源代码 bounce1d.c。

(2)编译运行 bounce1d.c。记录实验结果。

程序运行时，输入以下几种情况，记录实验结果。

(a) 按下 3 次 f 键。

(b) 按下 3 次 s 键。

(c) 按下 3 次空格键。

(d) 按下 Q 键。

在(a)、(b)、(c)三种情况中，写出初始间隔值和重复间隔值的变化情况(如果有变化)。

5.在 1-4 的 基础上，自行设计完成一个功能完整的视频游戏

## 程序编写要求：

1. 程序开始几行应为对应的源程序说明， 源程序中的重要语句要有注释 ；
2. 用到 curses 库函数、间隔计时器、信号处理