Escribir código de Xen es, en muchos sentidos, similar a la escritura de código UNIX.

El siguiente ejemplo muestra algunos de los puntos a los cuales nos estamos refiriendo, siendo el primer bloque correspondiente a UNIX y el segundo bloque correspondiente a XEN:

System Calls - Hypercalls
Signals - Events
Filesystem - XenStore
POSIX Shared - Memory Grant Tables

En un sistema UNIX, es posible configurar un handle para distintas señales. Se trata de una simple asincrónico
que permite que algo fuera del programa permita ofrecer un único bit de los datos diciendo "se ha producido un evento x". Todo ello gestionado por el Kernel u otra aplicación.

En Xen la analogía es el mecanismo de los eventos. Una de las primeras cosas que un Kernel invitado debe hacer es registrarse, con una llamada a ser utilizado por otro eveto de entrega de servicio. Cuando los eventos son devueltos a los clientes, diferentes flags se establecen para indicar que está presente caso y que algo ha ocurrido. Si la entrega del evento contiene una marcara, estos parámetros pueden ser utilizados más tarde para comprobar si hay espera de algún tipo de mensaje. En caso contrario se entregará de forma asíncrona.

Los eventos de Xen pueden venir, y pueden representar ciertos elementos o interrupciones de hardware virtual (interrupts), o pueden ser planteados por otros indicadores. Al igual que ocurre con las señales de UNIX, pueden ser utilizados para construir más complejos elementos de comunicación asincrónica, por ejemplo, la asignación de un evento de canal para indicar que el contenido de una página de memoria compartida han sido actualizadas.

Al igual que una señal, Xen realiza una entrega a través de una llamada. Todos son "upcalls" entregados a la misma dirección. Esto es similar al mecanismo utilizado para UNIX en algunos elementos como el microkernel del sistema operativo POSIX , como GNU Hurd, cuando las

señales se entregan mensaje a través de un puerto, a la aplicación de usuario, que luego se encarga de la Vectorización mediante el uso de handles todo funciona de una forma coherente y correcta.

Los eventos de Xen comparten mucho en común con las interrupciones de hardware y con las interrputs de UNIX , pero son conceptualmente una relación más estrecha de  Puertos feficientes. Los eventos Xen  se entregan a través de los canales, nadie no puede recibir eventos de un ligar arbitrario de un arbitrario remitente. Cuando un proceso de UNIX quiere enviar una señal a otra, primero debe descubrir el proceso de identificación a distancia y, a continuación, simplemente tiene que utilizar el sistema Kill para matar el proceso que en ese momento está en uso.