

Tipo de artículo: Artículo corto
Temática: Software libre
Recibido: 20/2/2012 | Aceptado: 26/3/2012

Estaciones de trabajo multi-asiento

Multiseat workstations

Daniel Hernandez Bahr^{1*}, Yunier Soler Franco¹

¹ Geitel. Centro de Gestión de Información y Tecnologías. Universidad de las Ciencias Informáticas, Carretera a San Antonio de los Baños, km 2½, Torrens, Boyeros, La Habana, Cuba. CP.: 19370, Correo-e: {dbahr, ysoler}@uci.cu

*Autor para correspondencia

Resumen

Con el desarrollo actual de los dispositivos de hardware los sistemas operativos modernos ejecutan simultáneamente un gran número de operaciones sin disminuir su rendimiento. Sin embargo, la inmensa mayoría de los recursos del sistema permanecen inactivos y el por ciento restante podría ser aprovechado por otros usuarios, permitiendo incrementar la eficiencia del sistema. Este uso compartido puede lograrse implementando una “configuración multi-asiento”. El presente trabajo describe los procedimientos necesarios para convertir una estación de trabajo convencional en una estación multi-asiento, recoge también las principales soluciones existentes y la integración del sistema operativo Nova con este tipo puestos de trabajo.

Palabras clave: GNU/Linux nova, PC multi-asiento, recursos de hardware, sistema X window, software libre.

Abstract

With the current development of hardware devices, modern operating systems are able to simultaneously execute a great number of operations without losing performance. However, most of the system resources will remain idle. That remaining percent could be used by other users, allowing increasing system's efficiency. This shared use can be achieved by implementing a “multi-seat configuration”. This paper describes the procedures needed in order to turn a conventional workstation into a multi-seat workstation; it covers also the main existent solutions and the integration of the Nova operating system with this kind of workstations.

Keywords: Free software, hardware resources, multiseat PC, nova GNU/Linux, X window system.

1. Introducción

Las primeras computadoras eran equipos enormes con capacidades computacionales extremadamente lentas y muy complejas de operar. Por lo general un técnico entraba manualmente, desde un panel de control, pequeños programas que normalmente demoraban horas, a veces incluso semanas en ejecutarse. Con el transcurso de los años las máquinas fueron ganando en velocidad de procesamiento, lo que disminuyó el tiempo de corrida de los programas, haciendo que la demora entre el fin de la ejecución de un programa y el inicio del siguiente se convirtiera en un aspecto a considerar. Pronto, las metodologías de procesamiento por lotes evolucionaron para disminuir estos tiempos muertos, poniendo los programas en colas de manera que, tan pronto uno terminara, el siguiente comenzaría.

Alrededor de 1955 John McCarthy pensaba en un sistema de tiempo compartido, refiriéndose a un sistema operativo que permitiera a cada usuario de una computadora comportarse como si estuviera en total control de la computadora, no necesariamente idéntica a la máquina en la que corre el sistema operativo. A decir verdad Christopher Strachey había comentado a Donald Knuth que el término tiempo-compartido (del inglés, *time-sharing*) ya estaba en uso para referirse al tiempo-compartido por los programas escritos para ejecutarse simultáneamente, idea que se encontraba implementada en el sistema SAGE (McCarthy, 1983).

El concepto de tiempo-compartido fue desarrollado a partir del criterio de que, mientras un solo usuario resultaba ineficiente, un gran grupo de usuarios trabajando juntos no lo era. Según describe el patrón de interacción, en la mayor parte de los casos los usuarios entraban relativamente grandes volúmenes de información seguidos por una larga pausa, pero un grupo grande de usuarios trabajando al mismo tiempo significaría que las pausas para un usuario se utilizarían para la actividad de los otros. Teniendo un grupo de tamaño óptimo, el proceso general resultaría muy eficiente.

El concepto fue descrito por primera vez públicamente en 1957 por Bob Bemer (Bemer, 1957) y el primer proyecto en implementarlo fue dirigido por John McCarthy en 1957 en el Instituto de Tecnologías de Massachussets (MIT, por sus siglas en inglés), sobre una IBM 704 modificada primero y luego una IBM 7090 (McCarthy, 1983). El resultado del proyecto fue el Sistema de Tiempo Compartido Compatible (del inglés, *Compatible Time-Sharing System*) (CTSS, por sus siglas en inglés). Otros sistemas que implementaron el paradigma de tiempo-compartido fueron PLATO II (desarrollado en la Universidad de Illinois) y *Dartmouth Time Sharing System* (primer sistema de tiempo compartido comercialmente exitoso).

En la década de los 70 encontrar un escenario en el que varias terminales (incluso gráficas) se conectaban mediante módem, puerto serie RS-232 o directamente a un mainframe era muy común. En muchas implementaciones el mainframe encuestaba secuencialmente a las terminales buscando datos adicionales o alguna acción solicitada por el usuario. Posteriormente las interconexiones eran guiadas por interrupciones y algunas utilizaban tecnologías de transferencia de datos en paralelo, como el estándar IEEE 488.

Con la invención de las redes basadas en el Protocolo Internet (*IP* por sus siglas en inglés) resultaba muy sencillo para los usuarios iniciar sesiones utilizando telnet, para sesiones de interfaz de línea de comandos (CLI, por sus siglas en inglés), o algún servidor “X Window” para entornos gráficos.

Con el auge de los microprocesadores los sistemas de tiempo compartido desaparecieron poco a poco, ya que los microprocesadores resultaban lo suficientemente eficientes y baratos como para que una sola persona pudiera disponer de todo el tiempo de procesamiento dedicado a sus necesidades, incluso cuando este se encontraba inactivo.

2. Desarrollo

2.1 Sistema X Window

El protocolo X Window (también conocido como X11, por ser la 11 su versión actual, o simplemente X) provee las bases para desplegar interfaces gráficas de usuario (GUI, por sus siglas en inglés) y capacidades de entrada de datos mediante dispositivos que permiten a varios usuarios explotar el poder de procesamiento de sistemas de tiempo compartido.

X utiliza un modelo cliente – servidor: un servidor X se comunica con varios programas cliente. El servidor recibe solicitudes de salida gráfica (ventanas) y retro-alimenta con datos de entrada del usuario (teclado, ratón, pantallas

táctiles). Esta terminología (el terminal del usuario siendo el servidor y las aplicaciones, los clientes) a menudo confunde a los usuarios de X, debido a que los términos parecen estar invertidos. El problema reside en que X asume el punto de vista de la aplicación y no el del usuario final: X provee servicios de visualización y entrada/salida de datos para las aplicaciones (por eso es el servidor) y las aplicaciones consumen estos servicios (por eso son clientes).

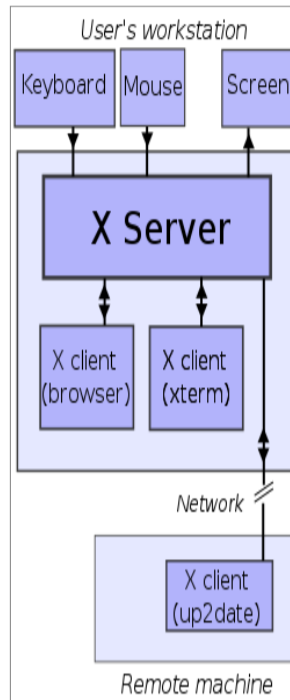


Figura 1. Ejemplo de despliegue de un servidor X.

El protocolo de comunicación entre clientes y servidor opera transparente a la red, el cliente y el servidor pueden ejecutarse en la misma máquina o en computadoras diferentes, posiblemente con diferentes arquitecturas y sistemas operativos. Incluso un cliente puede emular un servidor X y proveer servicios de visualización a otros clientes. Esto es conocido como “X anidado” (del inglés, *X nesting*).

2.2 Estaciones de trabajo Multiusuario

La primera implementación de una interfaz X Window con soporte para múltiples consolas en una misma PC fue desarrollada por Miguel Freitas en 2001, utilizando un sistema operativo con núcleo Linux y el sistema gráfico X11 (en aquel momento mantenido por XFree86 (Freitas, 2005)). La solución consistía en modificar el servidor X para que permitiera la ejecución de varias instancias concurrentemente, de manera que cada una capturara eventos de un ratón y teclado e imprimiera el contenido gráfico en un monitor específico. Este método fue denominado “multi-asiento” o “multi-terminal”.

En 2002 una compañía canadiense, *Userful Corporation*, liberó una solución para gestionar sistemas multi-asientos sobre GNU/Linux llamada *Userful Multiplier*, que permite hasta a 10 usuarios compartir una computadora simultáneamente. *Userful Corporation* ha liberado diferentes versiones de la solución, como la 3.7 en 2010 (Cox, 2010). Actualmente se encuentra en la versión 4.0, compatible con Ubuntu 10.04 y otras distribuciones de GNU/Linux. Otras soluciones surgieron en el año 2003, como la presentada por Svetoslav Slavtchev, Aivil Stoss y James Simmons, cuyo acercamiento consiste en una modificación del kernel Linux, en su rama 2.4, denominada *Backstreet Ruby* (Slavtchev, 2003) utilizando *evdev* y *Faketty*.

El Centro para la Computación Científica y el Software Libre (C3SL) de la Universidad Federal de Paraná, en Brasil, creó en 2005 una solución pasada en servidores X anidados, como *Xnest* y *Xephyr*. En 2008, el centro liberó el *Multiseat Display Manager* (Gestor de Display Multi-asiento) para facilitar el proceso de instalación y configuración.

2.3 Multiseat Display Manager

Se trata de una herramienta para configurar varias estaciones de trabajo en una misma computadora a partir de varias salidas de video y dispositivos de entrada. Su funcionamiento se puede describir en cuatro pasos fundamentales:

1. Detección de tarjetas de video.
2. Detección de dispositivos de entrada.
3. Conformación de las estaciones de trabajo.
4. Configuración del Servidor X.

2.3.1 Detección de tarjetas de video

Utilizando la salida del comando *lspci* para puerto VGA se obtienen los identificadores de los buses asociados a los dispositivos.

```
$ lspci | grep VGA | cut -d " " -f1 | tac  
  
00:0a.0  
  
00:09.0
```

A partir de esta información es posible determinar otros datos como el controlador que utilizan y los modos de vídeo que soportan. La herramienta *xrandr* brinda información sobre los modos soportados por cada salida de vídeo conectada.

2.3.2 Detección de dispositivos de entrada

Mediante el uso de la funcionalidad de *hal-find-by-capability* es posible conocer los objetos de los dispositivos que se soliciten de la base de datos de *HAL*, por ejemplo:

```
$ hal-find-by-capability --capability input.mouse  
  
/org/freedesktop/Hal/devices/platform_i8042_i8042_AUX3_port_logicaldev_input  
  
/org/freedesktop/Hal/devices/usb_device_46d_c018_noserial_if0_logicaldev_input  
  
$ hal-find-by-capability --capability input.keyboard  
  
/org/freedesktop/Hal/devices/platform_i8042_i8042_KBD_port_logicaldev_input
```

De esta información se pueden obtener datos de los dispositivos que son de interés para la configuración de la estación de trabajo, por ejemplo:

```
$ EVDEV_NODE=$(hal-device  
/org/freedesktop/Hal/devices/platform_i8042_i8042_KBD_port_logicaldev_input | grep linux.device_file |  
cut -d'"' -f2)  
  
$ echo $EVDEV_NODE  
  
/dev/input/event3
```

2.3.3 Conformación de una estación de trabajo

Conociendo los dispositivos conectados se procede a conformar las estaciones de trabajo asociando un teclado y un ratón a cada monitor. Para esto se envía un mensaje a cada monitor que solicita presionar una tecla del teclado y un botón del ratón, quedando únicamente configurar el Servidor X con toda la información recopilada.

2.3.4 Configuración del Servidor X

La configuración del Servidor X se realiza en el archivo *xorg.conf* donde se definen varias secciones como se listan a continuación:

⤴ **InputDevice:** Se describen los dispositivos de entrada conectados al equipo. A cada dispositivo se le asigna un identificador, además de definirse opciones de configuración para cada uno.

```
Section "InputDevice"
    Identifier      "Keyboard1"
    Option "Device"    "/dev/input/event3"
    ...
EndSection
...
Section "InputDevice"
    Identifier      "Mouse1"
    Option "Device"    "/dev/input/event8"
    ...
EndSection
```

⤴ **Monitor:** Se describen los datos de cada monitor especificándose además el identificador de cada dispositivo.

```
Section "Monitor"
    Identifier      "monitor1"
    ...
EndSection
```

⤴ **Device:** Se definen las configuraciones de las tarjetas de video conectadas, siendo requeridos los datos de un identificador y un controlador para cada dispositivo.

```
Section "Device"
    Identifier      "device1"
    BusID "PCI: 0:9:0"
    ...
EndSection
```

⤴ **Screen:** Se asocian en esta sección los identificadores de los monitores y las tarjetas de video de finidos en las secciones **Monitor** y **Device** respectivamente.

```
Section "Screen"
    Identifier      "screen1"
    Device "device1"
    Monitor "monitor1"
    ...
EndSection
```

⤴ **ServerLayout:** Le son asignados a cada sección **Screen** los dispositivos de entrada correspondientes a través de su identificador de la sección **InputDevice**.

```
Section "ServerLayout"
    Identifier      "seat1"
    Screen          0 "screen1" 0 0
    InputDevice    "Mouse1" "CorePointer"
    InputDevice    "Keyboard1" "CoreKeyboard"
EndSection
```

De este modo quedan configuradas las estaciones de trabajo para todos los dispositivos conectados. Este proceso se ejecuta automáticamente al iniciarse el sistema.

2.4 Estudio de Caso: Proyecto Paraná Digital

Proyecto de inclusión digital para las escuelas públicas del estado de Paraná, Brasil. Consiste en la creación de laboratorios de computación multi-asiento en 2100 escuelas públicas. Cada laboratorio cuenta con 15 equipos de 4 asientos (monitor, teclado y ratón) cada uno, utilizando Debian GNU/Linux como sistema operativo. El costo por concepto de hardware es un 50% menor. En total se encuentran instaladas 44000 computadoras de las que se benefician 1500000 alumnos y 57000 profesores.

Aunque el proyecto no ha terminado, los resultados con la solución son muy prometedores.

2.5 Entorno nacional

La empresa GEDEME se encuentra en proceso de ensamblaje de un número de estaciones de trabajo multi-asiento en aras de brindar una gama más amplia de opciones a las necesidades del programa de informatización de la sociedad. En entornos relativamente pequeños, una solución de clientes ligeros resulta más costosa por concepto de instalación, mantenimiento y administración, puesto que es necesario adquirir las estaciones de trabajo, el servidor, el cableado de la red, la climatización y los especialistas con entrenamiento para administrar el sistema.

Los equipos ensamblados por GEDEME dispondrán de un sistema operativo NOVA configurado para funcionar con múltiples asientos.

3. Conclusiones

Implementar un sistema multi-asiento para entornos de trabajo de pocos puestos resulta no solo más económico sino además más fácil de controlar, ya que se trata simplemente de un sistema operativo con varios usuarios.

La solución conjunta del hardware ensamblado por GEDEME y el Sistema Operativo provisto por Nova constituyen un paso de avance importante en el proceso de informatización de la sociedad puesto que, al reducir costos de adquisición y mantenimiento para laboratorios computacionales relativamente pequeños, permiten un mayor acceso a esta tecnología.

4. Referencias

BEMER, B. *How to consider a computer*, Automatic Control Magazine, Data Control Section, 1957: 66-69.

COX, M. *Useful upgrades multi-seat Linux desktop virtualization solution*, 2010. Disponible en: [\[http://www.echannelline.com/usa/story.cfm?item=25455\]](http://www.echannelline.com/usa/story.cfm?item=25455).

FREITAS, M. *Multiple local XFree users under Linux*, 2005. Disponible en [\[http://cambuca.ldhs.cetuc.puc-rio.br/multiuser/\]](http://cambuca.ldhs.cetuc.puc-rio.br/multiuser/).

MCCARTHY, J. *Reminiscences on the History of Time Sharing*, 1983. Disponible en: [<http://www-formal.stanford.edu/jmc/history/timesharing/timesharing.html>].

SLAVTCHEV, S. *XFree Local Multi-User HOWTO*, 2003. Disponible en: [<http://www.tldp.org/HOWTO/XFree-Local-multi-user-HOWTO/>].