

Memoria de la puesta a punto de valticall1 para su uso como servidor del SBC v0.1

Esteban Fullana

October 21, 2003

Abstract

Perdonad que lo haga tipo memoria pero a ver si os sirve de utilidad.

1 Introduccion

Tical11 tiene varias funcionalidades:

- Es el servidor de sistema del SBC. Recordemos que el SBC no tiene disco duro, por tanto cuando arranca importa su kernel por sftp de tical11. Además todo el sistema de archivos del SBC es importado por NFS tambien desde tical11.
- Es la salida al exterior del SBC. Todo el sistema está ideado para que sea lo más transportable posible. Por eso Javi y yo decidimos que el SBC se comunicara con tical11 por una red interna. A su vez aprovechando sus dos entradas de red, tical11 se conecta con la red exterior por la entrada de red libre.
- También sirve como ROB emulator.

2 El arranque

Una vez llegó tical11 se instaló una versión de Red Hat y se le empezó a preparar como servidor del SBC. Para ello lo primero que hubo que hacer es compilar un kernel especial para arrancar sistemas sin Discos Duros y que iba a ser importado por el SBC. Hacer la compilación del kernel fue bastante fácil y que el SBC arrancara con el también; pero hay que decir aquí que el SBC debe ser configurado en su BIOS para que arranque desde internet

(ahora mismo ya esta configurado, pero si se quita la batería habrá que volver a configurarlo). Para ver el arranque del SBC debes conectar un cable desde la salida term del SBC al puerto serie de valtical11 y utilizar la aplicación minicom. Tampoco es trivial acceder a la BIOS del SBC. Construimos Javi y yo un teclado especial solo para hacer eso (esta en el armario de Valencia del 40, el blanco), pero eso Markus Joos os lo puede hacer con mucha facilidad sin necesidad de teclados. De todos modos solo con el traslado, si no habeis tocado nada todo debería funcionar igual que en el testbeam. Hay que decir aqui q es posible que SBC se conecte directamente a la red del CERN y que arranque de ahi. Markus trabajo en eso, si lo quereis cambiar estoy seguro que Markus os ayudaría. Una vez arrancado tuvimos un problema y es que el SBC no incorpora AFS. Esto supone un problema para instalar el software del TDAQ. Esto implica que el software debe instalarse en valtical11, en la parte de sus archivos que se importa por NFS al SBC. Este hecho hace que la compilación tenga que hacerse en valtical11, lo cual no supone ningun problema para las aplicaciones para se ejecutan en ese ordenador (filarscope...etc) pero si lo es para las aplicaciones q se ejecutan en el SBC, como por ejemplo todo lo relacionado con VME, muy especialmente los drivers. El problema es que unos drivers compilados en un kernel no sirven para otro kernel y como el SBC usaba un kernel especial los drivers no se podían cargar. La solución fue cambiar el kernel de ticall1 para que al compilar los driver de VME (y otros) en ese ordenador fueran compatibles con el kernel del SBC. Por eso valtical11 tiene dos kernels el de por defecto fue el definitivo y el otro lo conservamos por si acaso.

3 El software

El software del tdaq esta en dos lugares, una esta en /work_local/tdaq/DataFlow y la otra esta en /home/lfullana/DF. La primera debía ser la versión oficial del ordenador y la última me la instalé yo para hacer pruebas con ella. La cuestión es que al final, como suele suceder, las pruebas salieron bien y no nos preocupamos de pasarlas a la versión oficial. El SW del tdaq tiene la organización de un sistema de SW mantenido por CMT: todo esta organizado en packages, algunos nos han sido útiles como el de la filar o el del VME, nos son útiles como el del TTCvi y otros nos serán utiles como el Rod Crate controller. Cada package tiene varios subdirectorios el directorio CMT es el más importante por dentro esta un archivo llamado requirements, donde hay que indicar como es la estructura del package:

- Cuantos ejecutables hay y cuales son.

- Que librerías este package pone a disposición de los otros packages.
- Que librerías de los otros packages necesita..

En concreto lo que hice yo fue crear un nuevo package, llamado `rod_rcc` que usaba como librerías de otros packages el del `vme_rcc` (que es el de configuración, lectura y escritura en el bus VME), y ponía a disposición de los otros packages las librerías del rodemo. Esta librerías del RODdemo son solo una forma comoda de configurar el ROD. En el manual del RoD esta donde y que tienes que escribir (o leer) en el bus VME para configurar el RoD, por ejemplo, para leer el registro del status del trigger hay que leer en la dirección `0x388001A2` (por ejemplo) los bits 12-14, bueno pues en lugar de hacer esto a pelohay una librería que es `ReadTrigStatus` que ya se encarga de leer donde toca. Estas son las librerías que hay en el `rod_rcc`. Pero claro estan librerías necesitan de una librería más profunda que lean y escriban en el bus VME, yo lo que hice fue enlazar estas con el package `vme_rcc` que ya hacía esto. Por tanto todas las librerías del RoDDemo estan a disposición de todos los packages.

4 El TTCpr

Unos de los paquetes del software del tdaq es el llamado `ttcpr`, este paquete tenía una aplicación que te volcaba en pantalla la información del trigger que le llegaba al `ttcpr`. Como había una librería del RoDDemo que se encargaba de escribir la información del trigger en el ROD por VME, lo único que tuve que hacer es modificar la aplicación y que en lugar de volcarla por pantalla que llamara a la librería del RoDLib y la volcara en el bus VME.