

Abstract

Most Linux based z/Server shops already have a decent amount of experience with Linux, however, many do not have experience with the z/VM operating environment. This is why we propose an open source z/VM Graphical User Interface for the IBM Linux Challenge project. We feel that a powerful and easy to use GUI for z/VM will allow z/VM administrators of all experience levels to complete z/VM related tasks (including tasks related to hosting Linux environments) quickly and in a more efficient and user friendly manner. This will help speed up adoption of Linux on z/Series because businesses will not have to fight the z/VM learning curve, instead they can dive right into using Linux on z/Series.

Introduction

The goals of this project are to design a z/VM graphical user interface that meets the following requirements. It must perform Administrative tasks in z/VM from a Graphical User Interface. Little or no CP/CMS command knowledge necessary to operate. Interface should be self documented. Portable Program Operation supporting many platforms, including Linux.

We wish to minimize z/VM's complexity by providing a graphical user interface through which an administrator can interact with z/VM to perform routine administrative tasks. An intuitive graphical user interface will allow the administrator to save time by not having to type (or memorize) the many various (and sometimes lengthy) z/VM commands that are needed to properly manage the hosted virtual environment.

To the novice z/VM user, much of z/VM can seem unfamiliar and even unnecessary. Many z/VM commands have a lot of extra options that can be confusing and often require nothing more than a sane default value. In the design of the GUI, care should be taken to ensure that very little CP/CMS command knowledge is necessary to operate the GUI and perform administrative tasks with z/VM.

Although we will attempt to reduce the amount of complexity the end user must face in order to administer z/VM, it is impossible to remove all of the complexity involved. There are certain tasks and terms or acronyms that the user will need to understand in order to be sure that he/she knows what is happening within z/VM. In such cases the interface should be documented through the use of tool-tips, in-line help buttons or menus, or through other intuitive, easy and non-intrusive methods. Examples of such cases include acronyms like OSA, QDIO, DASD and CP (Central Processor).

The Java programming language will be used to ensure that the GUI runs on many operating systems as possible,

Device management

The user will have the ability to attach and detach all devices to/from guests. The user may also view status for all devices both free and used.

The z/VM GUI Project will work with both types (virtual and real) of DASD devices where it is appropriate to do so. The user will have the ability to attach and detach DASD volumes to/from SYSTEM and guests on the system.

Before a DASD device can be carved up into minidisks or used by z/VM it must be formatted. Through use of CPFMTXA the user may format DASD packs of all allocation types: PAGE, PERM, SPOL, or TDSK.

The z/VM GUI Project will also support a timed format/copy queue which can be loaded with DASD format or copy jobs that can be batched and set to run at a specific time of day. This allows an administrator to batch queue a large number of jobs so he/she does not have to wait around to start another job after the completion of the current job. A specific time may be set to allow formatting/copying to be started during times when the system will be under minimum load. This will minimize the impact that larger jobs will have on other users of the system. The queue will allow each DASD to have its own allocation type and volume label. It will also support the auto incrementing of volume labels during batch job creation. This saves time as the administrator will not have to repetitively type many volume labels that are similar.

Guest management

Managing guests is the most important and time consuming task that is performed by a z/VM administrator. Guests must be created with a user name, a password, and a set of devices that can be accessed and used in various ways by the guest. It may also be necessary to either extend or revoke privileges to certain devices for a particular guest. When a guest is no longer needed, it must also be deleted and its resources reclaimed.

The z/VM GUI Program will allow creation, modification and deletion of guest's directory statements. All supported tasks will be performed graphically.

The user may completely remove a guest from the user directory. This removes all directory statements associated with the guest. Optionally, the user may choose to format the guest's DASD packs to prevent someone else from recovering it after the pack is reassigned to another user.

The user will have the ability to manage the priority of system resources. Through the use of QUICKDSP (quick dispatch) and z/VM's ability to prevent a guest from using more than a specified amount of CPU time and I/O bandwidth, guests access to system resource can be moderated.

Live System Status

The user will see a system summary that will give him/her a sense of how the system and its guests are behaving. This includes real time monitoring of free storage, used storage, free and used page space and spool space, the number of guests logged on (and their names), and CPU utilization.

Alert conditions may be set and monitored. An alert condition is a "statistic / condition" pair that can be used to perform an action that will alert an administrator if the system needs attention. For example, an administrator

could monitor storage (the statistic) for the condition that if the amount free space drops below 10% then an action should be performed. The type of action that will be supported will be the execution of an external program. In a Linux environment, this external program could be a shell script that is set to send an email to an address that resolves to a cell phone. The end result is that the administrator will receive a phone call from his overly stressed out z/Server.

System Management

The user should have the ability to quickly "ban" a guest from the system that he/she deems a threat to system security. The guest in question will be instantly forced off of the system and the guests password will optionally be changed to "NOLOG" to prevent that guest from restarting until the administrator lifts the ban.

The user may add/remove VM page/spool packs. These additions will be placed in the SYSTEM CONFIG to make them permanent. The administrator should be able to see how many slots are open for CP OWNED packs.

Through the Q SRM and SET SRM commands, the user may view and set values for the Q1, Q2, Q3 and Q4 dispatch Queues.

Guest LAN / vswitch Management

The user may create and delete Guest Lans. and vswitches. Lans/vswitches that are created can be optionally entered in the SYSTEM CONFIG to make their presence permanent. If a restricted guest LAN/vswitch has been created, an administrator should be able to restrict access to it.

Implementation Details

The z/Vm GUI project has been an ongoing research and development endeavor at Clarkson university. We have already designed two custom methods to communicate with z/VM. We have also created two useful utilities, an active CMS User Monitor and a graphical CMS File Editor.

We are able to easily communicate with z/VM through the use of a custom designed Telnet module that allows us to execute z/VM commands and receive command output. This is a very powerful method of communication as arbitrary command execution is possible and no extra software is needed of the z/VM system. We also have a Systems Management API communication module in development which will simplify guest management,

Using these communication mechanisms, we have developed a utility that will show an administrator who is logged on to the system. The administrator can force users off the system and log on (XAUTOLOG) users if desired.

We have also developed a graphical CMS file editor that allows a user to open any specified file on any specified disk of any guest on the system. A user friendly file editor window appears and the user can end the file using the standard home, end, page up/down, copy, paste, and save functionality instead of having to learn how to use XEDIT.