Our Users Have Root!

Laura de Leon, Mike Rodriquez, & Brent Thompson
Hewlett-Packard Company

Our Users Have Root!

Laura de Leon, Mike Rodriquez, & Brent Thompson – Hewlett-Packard Company

ABSTRACT

This paper describes how things work at our site, where users have responsibility for administering their own workstations.

Hewlett-Packard Laboratories is HP's primary advanced R&D laboratory, with about 1000 people and about 1500 computer systems. A central support organization within HP Labs is charged with creating an infrastructure to enable this to work successfully, even though they do not have root access on most systems in the division.

The paper gives some examples of how things are done differently than at other sites, and details what is in place. It also gives some areas where there is work yet to be done.

Introduction

Our users have root! In fact, our users have total responsibility for administering their own workstations. In most cases, we as a support organization do not have user or root accounts on the systems. We attempt to provide the support and infrastructure necessary for this model to work for everyone involved.

This paper is not going to justify why users are responsible for their workstations – it is a fact of life for us, and good or bad, it is not going to change soon.

This situation does have a major impact on what we do and how we do it. This paper will address how we handle things differently than we would if we controlled all (or most) of the environment here. It will identify some issues that we have not handled yet.

Our Site

Hewlett-Packard Laboratories (HP Labs) is HP's primary advanced research laboratory. The Palo Alto site has about 1000 people, with about 900 Unix systems and about 600 PCs scattered through 6 buildings separated by up to two miles. These people are our customers. Most of the people with Unix workstations are the researchers and their support staff.

Most of our computer systems are made by one vendor (HP of course), but they are of several different flavors and types (often incompatible), running different versions of the OS. In addition, there are people doing operating system research running various other systems.

Research Computing Services (RCS) supports as much of this environment as possible. We provide the resources necessary for people to run their own machines.

The Challenge

Our customers want to spend as little time as possible administering their computer systems, they want an advanced working environment with the tools they need to do their jobs, they want to spend as little money as possible, and they want complete flexibility. They do not have to use any particular solution we present, if they perceive it as not meeting their needs or as too much trouble.

We try very hard to satisfy all these, including flexibility. We want maintainable solutions, based as much as possible on a common environment we can convince everyone to use – the environment must be as supportable as possible. We want to help everyone make reasonable use of their resources. We want to enable everyone to do their jobs more efficiently.

Our responsibilities include providing services such as e-mail, news, printing, application support, OS updates, and file sharing, and providing support for the customers when they have difficulty with their workstations, whether as users or as administrators.

Some of the challenges we face, that are different than they would be if we had central control:

- System Integrity – we can't depend that ANYTHING is the way it should be on any system we are asked to help fix.
- Inconsistency – For example, we can't do site hiding because users' names on their workstations may not be the names registered on the mail server.
- User error – having to help users solve problems they have gotten themselves into (the 'rm -r /' syndrome).
- Troubleshooting – we are often called upon to troubleshoot a problem on systems to which we have no access.
- Security – We don't control systems, we don't know how secure they are.

- Not being able to do it ourselves – some things might be easier in batch (like OS updates) but we can't make that decision.
- Sometimes, users don't know what is important – things like running regular backups.

We try to address these problems by providing tools, infrastructure, and skilled support staff. We can make these available, but we can't force customers to install or use them, even if things work better when they do.

Some Things Don't Change

We offer news via NNTP, and people can request an account on a timeshare to read news. This is probably how we'd do news even if we controlled the workstations.

How we handle the network also doesn't change much, at least partially because people usually act responsibly (and systems often break right away if users don't follow network rules). Handling of IP addresses is described a little further on.

Our Organization

Research Computing Services (RCS) has a Customer Support group, that does the front line support. Our customers are free to call on us for any computer related issue; they start by calling or sending mail to the Customer Support Service Desk. The Service Desk dispatches most of the calls to the front line support groups, and the remaining ones directly to the rest of RCS.

Front line support is actually two groups – the System Support Group does basic troubleshooting and On Site Support deals with hardware and other issues most appropriately handled physically at the system in question. The front line support people have a small but growing set of "Standard Answers" available from the rest of the organization – these are descriptions of common problems or situations, and the official way of dealing with them. If there is no Standard Answer for the call, and they can't fairly quickly create an ad-hoc diagnosis, then the call goes to the "technology owner" – a member of another part of RCS.

The other groups in RCS are Network Engineering, Applications, Client Platform, and Site Platform. Every individual in these groups is responsible for some products or areas (the "technology owner"). If no individual has responsibility for a product or gray area, the appropriate manager is responsible. Managers can deal with the question or problem themselves, or assign it to someone in their group. The official level of support for any particular item is allowed to be "none".

Customers can call with any sort of problem and we will help them if possible. This sometimes requires the support person having access to the system. Customers have the choice of giving access or not. It's all up to them. No access pretty much precludes us helping solve the problem. When access is granted, it has ranged everywhere from just reporting by phone the results of what the customer thinks the support person said to type, to giving the support person the root password.

The expectation is that we will troubleshoot, and help the customer implement a solution to their problem. In practice, we often end up fixing things for them, leaving them none the wiser about how things work (at the customer's choice).

We are writing this paper from the point of view of the Site Platform group, which is responsible for the site infrastructure – servers and services that affect the entire site (other than physical networking).

Actually, We Do Have Root Sometimes

Well, yes, we do have root capability on many machines, usually once a day. How this happens is: we have convinced many/most of our users to install a service we provide to "keep important files up-to-date". This service is called "Ninstall Star".

Ninstall is an internally developed client/server network software distribution utility widely used throughout HP. Ninstall(1L) is the client program; it installs software onto the client system according to the installation specifications contained in "packages" on the server. One important feature of ninstall is that it includes a preview option, to see what would be done (installed, updated, deleted, chmod'd, touch'd, etc.) if the package were actually ninstalled. (Ninstall was described in a paper presented at the first LISA conference in 1987, and also a more detailed paper presented at Uniforum, 8-11 February 1988, pp. 41-53.)

What we have done is to encourage HP-UX workstation owners to frequently install every available package from a particular one of our ninstall servers, i.e.:

```
ninstall -v -h hpllan "*"
```

Most users have cron(1) do it nightly.

Some users choose not to run Ninstall Star from cron. These folks usually know what they are doing, and just prefer to occasionally execute this command manually. Some users don't install "*" but instead select for themselves which of the available packages they want kept up-to-date on their systems.

The ninstall preview option allows users to verify for themselves whether any ninstall package is acceptable to install, and as a rule packages should be previewed before installation. Ninstall Star from cron requires users to trust us enough to run it without previewing.

Even though we COULD do anything as root we want on every system that uses this service (but probably only once!), in fact there are many constraints. An implication of being granted root like this for a few moments each day is that we must be very careful how we use this capability. Once the trust given us is lost, it would be very difficult to regain.

Users expect Ninstall Star to not change the behavior of their systems. The purpose of Ninstall Star is to keep up-to-date those files that RCS has determined need to be upgraded independent of OS upgrades. Primarily this includes files that are volatile in nature, or are "infrastructure configuration files". Specific examples include: /etc/hosts (useful for comments contained therein), /etc/resolv.conf, and printer config files. Although we might occasionally update programs via Ninstall Star, this would only happen for "emergency" reasons, not simply because a new program is available.

An example of when it was felt necessary to update binary files was during the infamous Internet worm situation of November 1988. At the time, not all of our workstations were protected by firewall gateways. Our sendmail was secure from this attack, but we were vulnerable to the ftp daemon opening. So, we used Ninstall Star to correct ftpd. Given the seriousness of the situation, we received no complaints about this action.

Another anomalous use of Ninstall Star was for data gathering. We wanted to estimate the amount of disk space in use at HP Labs. So that folks wouldn't perceive us as "snooping", we structured this temporary package to be easily excludable and announced our intentions well in advance, including details of the steps necessary to prevent the disksize script from running or to run it by itself. This also allowed the "non-cron" crowd to participate in the measurement. We then let Ninstall Star run the disk size program for about a week. Again, we had no complaints, and received feedback that people were pleased we had informed them in advance.

We don't handle configuring printers on workstations much differently than we would if we "really had root" – except of course that users choose for themselves whether to keep their list of configured printers up-to-date via Ninstall Star or not.

Currently, Ninstall Star configures all public printers in the buildings nearest to each client system. This means that when a new public printer is deployed, users on systems running Ninstall Star automatically have access to it. As the number of printers is growing, we are evaluating methods to allow users to easily select a subset of all the available printers.

Self-Actualized OS Update

Customers can do it themselves. In fact, they really must do it themselves. Again, this revolves around that major feature of our environment: that our users have root capability on their systems . . . and we do not. Whether and when an OS update occurs is purely up to each individual user. Because it is so much faster, simpler, and less dependent on peripherals any given system may not have (e.g., tape drives), we do all our OS updates over the network. The OS update procedure we provide must be a pull operation, not a push.

Whenever a customer decides to do it – she does it! (In order to maintain a manageable load on the servers, we do limit the number of concurrent update sessions – this limit is high enough it has never been reached yet, but could interfere with a user's preferred schedule.)

One result of users doing OS updates on their own schedules is that we must continue providing each version of the OS almost indefinitely. We basically support only HP computers, which simplifies our lives a bit. Even so, we support three types of hardware running Unix (HP-UX), and currently provide up to four versions of HP-UX for each of these architectures (meaning major OS revisions, the oldest released as long as four years ago).

Also, we always provide HP-UX in four major flavors (combinations of these two option sets): with HP Labs customizations vs. pure vanilla product; and all local-disk-resident vs. approximately 50% offloaded via symlinks to our central server (we call this "nfs-linked").

The algorithm for deciding whether we include any given file in the list to be nfs-linked is based on frequency/importance of use. We nfs-link only "unimportant" files, i.e., files that are not normally used for bootup, not normally used during login, not important for system diagnosis/reconfiguration/fixup, not used during the execution of normal activities, and not even accessed at all most of the time by most users. The file sometimes considered the archetypal member of this set is /usr/bin/banner – a file that is useful when you want it, but infrequently wanted at all and highly unlikely to be mission-critical when you do want it.

The selection algorithm is fairly stringent, which is why we only offload about 50% of HP-UX – for almost any particular system it's easy to find plenty more candidates. Even 50% or so of the (default) OS is quite a bit of space, though – who wouldn't appreciate an instant extra 40 MB of disk space with no hassle and no loss of functionality? And sure enough, most of our users choose to install the "nfs-linked" option. The "local" option exists mainly for those users who remember "like it was yesterday" when the server died for three hours in 1989 :-), and for certain time-share, cluster, or departmental servers.

Even most "local" users agree that local copies of the standard man pages are a waste of space, though, so we never automatically include any man pages in an OS update. [Even so, we do provide packages to get them by update across the network.] We always provide formatted, uncompressed copies of all standard man pages on our two central servers.

Ninstall(1L) is the main tool all our update procedures use. As previously mentioned, this tool is a generalized network software distribution utility that allows greater flexibility than the HP-UX tool update(1m), which is tuned to some particular aspects of HP-UX updates.

We distribute pre-customized default config files with HP-UX updates, so the system boots and works ok after being updated, no matter how different the new OS version may be from the old one. The update process deletes config files that are unchanged from a previously installed version; it saves changed ones with a .OLD suffix and generates a message to check for any customizations that might need to be merged back into the active version.

One feature we have included in our configuration files involves the consolidation of all system-specific information into one place, /etc/SYSTEM.INFO. We modify configuration files and software that need some piece of system-specific information such as TZ or BUILDING to read SYSTEM.INFO rather than having that information hard-coded. Such local information is extracted using the tool developed for that purpose, getinfo(1L), e.g.:

```
/etc/getinfo TZ
```

This single point of administration is convenient for us, but especially important for our root-wielding users who aren't experienced system administrators.

Standardized Systems

One side-effect of our (default) OS update process is that the invoking client ends up standardized, however briefly, to our division's notion of a standard system. Whether the owner subsequently undoes any of these standardizations is beyond our control of course, but mostly they don't (since it's easier just to choose a non-default installation option and not install our customizations at all than to install the default then remove our setup afterwards).

Therefore, this OS update process obviously can be, and often is, used simply to bring new systems up to the standard customization, e.g., newly purchased systems or systems of persons or groups newly brought into our division. We make all our update processes idempotent – there is never any harm in rerunning/restarting any update.

Patches

Along with the HP-UX update procedure, we provide a single-command procedure for installing whatever patches our group (RCS) currently recommends for each version of HP-UX. This patch procedure is described in documentation mailed to each client during the update process.

Patches always come unexpectedly, with no warning or schedule. The idea behind this recommended-patches command is that it should be run by cron every week to keep the system up to date with any recommended patches that may have come along in the meantime. Practically speaking, it's pretty disruptive to have the system patched and rebooted without warning by cron (if a recommended patch involving new kernel libraries comes along some week).

As with most of our update procedures, we provide a preview option which tells what would be done if the update were actually performed. We recommend that users should run the preview of recommended-patches every week via cron. If something turns up some week, they can just install the current recommended patches at their convenience.

As with everything we provide, whether any user installs the recommended-patches or not is up to them. To make it easy for the inevitable users who want to take advantage of the functionality and simplicity of the recommended-patch procedure, but want a different list of patches installed, this process is split into two parts, like most pieces of our OS update process. First the tools and lists are installed, then the tools are run against the lists. This way any user may install the tools first and modify or replace the default lists, then proceed with the second step that will act on his own lists. But these steps are only for users who desire this special feature; by default, the whole process just happens in one continuous step after the user specifies the single initial command.

We provide an ninstall package for every available patch, so users can easily install any patch they wish; the recommended-patches package is just a wrapper to make installation of the most important set of patches so easy even our most naive users will be able to do it themselves.

Applications

We also provide various applications for installation from our servers, such as Emacs, Epoch, FrameMaker, Lotus 1-2-3, LaTeX, LAN Manager, etc. They are all available using the same interface as HP-UX updates; like OS updates, "nfs-linked" and "local" versions are available for all. As with OS updates, the nfs-linked version is also most popular for applications. For example, consider this extreme case: FrameBuilder "local" installation = 85 MB, "nfs-linked" installation = 5 KB.

During OS updates, we try to determine if newer versions of any applications exist and, if so, attempt to update them automatically. Again, as with all our services, users can pull a new application onto their systems whenever they wish.

Applications (and other versions of HP-UX itself) are available from sources other than the RCS servers, so a flip side of this coin exists, too. Users control their own systems and they do install applications from those other sources. Sometimes this causes files to get overwritten, even important configuration files that we have worked hard to set up just right. Sometimes this causes systems to break in new and unexpected ways! So another result of only system owners controlling their systems is that troubleshooting is more complicated for us – we can never take anything for granted about the system setup. We always have to check even the most obvious, standard things: we've had printers break because spool directories weren't owned by 'lp'.

Documentation

We always have to write documentation for users who are not "real" system administrators.

One of the hurdles we have that is no doubt different from most other places is that our update procedure must result in a working system regardless of whether the user reads the documentation or not.

A fact of life for us is that usually a large percentage of users will not read it, and unfortunately not because they already know what it deals with – our users are not the sort who attend LISA! One of the Laws of Nature we have discovered (as have many other groups like ours) is that the number of users who will read any piece of documentation is inversely proportional to the length of that document. And, for any operation as complex as a complete OS update, it takes an awful lot of documentation to impart all the information that might be important to different users – all the more so since our users are so diverse.

So, we try our best to ensure users actually read the documentation. One way is to pare documents down to only what is needed for that particular phase. A rarely attained, though always remembered, goal is to have nothing longer than one (60-line) page.

File Sharing

We have no global home directory sharing (or even semi-global). All system owners feel (and are!) free to arbitrarily add and delete users at their own whim, so we have always considered it out of the question to maintain a common UID space. We do keep our central systems synchronized, but only one group is using these as their master list. They send us mail with a list of information, and we assign them a UID. This service is available to everyone, but it is not commonly known, and is fairly awkward to use.

The lack of a global UID space certainly complicates providing a service like home directory sharing. So far there has been little interest in this since all users have their own offices, their own workstations in those offices, and no perceived need to share home directories.

Different work groups do occasionally experiment with these notions of file sharing or common uids, but so far none of these experiments has caught on and spread.

We do have a form of global file sharing. We have public disks with minimal security that everyone can mount via NFS and LAN Manager for read/write access. These are available to anyone in the division with a workstation or a networked PC.

This is useful for making some files available to a few people, but the lack of security makes it less useful than it might be. In addition, ownership information is meaningless because of the lack of global UID space – UID 1234 might map to user 'A' on one system, and user 'B' on another.

A shared disk works as an alternative to mailing a file to a large group of users, for transferring files from PC to workstation or back, for putting out files people may or may not be interested in, and so forth. For example, someone might send a message that says "file ABC describes the plan for XYZ. If you are interested, read it and send me your comments", rather than sending file ABC to the full list of people.

Such shared disks are not useful for permanent storage, moving your home environment around, etc. Old files are subject to removal from these disks.

Many groups also have their own departmental servers, but we generally have no hand in either setting up or administering these, i.e., they are just among the undifferentiated set of client systems on which we have no access.

DFS (OSF's Distributed File System) with Kerberos authentication might help us by providing workstations that don't currently trust each other a way to do a truer level of file sharing, with some degree of trust in the security of what is being shared. Kerberos authentication seems like it may make many services easier to provide.

To distribute software, share system files, and support the "nfs-links" previously described, we have duplicate read-only file servers that have complete copies of all HP-UX and application bits we support. Essentially all workstations mount these disks (the particular ones relevant to each client system's architecture and OS version, that is); these are the disks to which we "nfs-link" various system files to save local disk space.

E-Mail

E-mail is another area where we have been successful in putting together a plan that works.

We have two completely different e-mail systems in use – much of the non-technical staff uses HPDesk, a proprietary e-mail system that runs on HP's MPE systems. An unrelated group of people at a central IT (as "MIS" is now known) facility manages HPDesk and the gateway to and from it.

Of the people who get their mail on Unix, most have mail delivered to their workstations. Timeshare facilities are available if customers want to get their mail in a central location; these are most often taken advantage of by users with PCs on their desks. Sometimes customers use the timeshared mail facility so they can have their mail in a central location where it is backed up, and they don't have to worry about keeping up with the software (and the patches).

We maintain a central list of mail aliases for everyone in the division. firstname_lastname is valid for everyone. We also set up (register) a username alias for everyone that gets mail on Unix, has an account on a Unix timeshare system, or requests a username alias.

Traditionally at HP Labs, the registered mail alias has been the user's last name (with a first initial prepended if needed for uniqueness). We recently reevaluated this policy and found there was no good reason for it, and that the policy resulted in extra complexity – quite a few people who wanted a username other than their last name were using their username of choice on their workstation, and setting up a local alias for the registered alias. This complexity works just fine, but it is nice from a global perspective to have the name a person sends from match her registered (user@hpl.hp.com) address.

We also maintain a number of mailing lists. We have a set of nested departmental mailing lists, so that anyone needing to send mail to a department or organization can send mail to the lists we maintain. The top level lists are moderated to avoid junk mail to everyone.

We periodically check these lists against departmental organization charts. Some departments provide us charts more often than others. Those organizations have more up-to-date lists. When we check our mailing lists, we find out if there is anyone we don't already have a firstname_lastname alias for. In practice, it seems that people get themselves signed up fairly promptly.

Our standard workstation sendmail.cf treats any mail for names not on the system (in either password or aliases file) as user@hpl.hp.com, and sends it to the mail hubs to be resolved. This sendmail.cf is among the files kept up-to-date by Ninstall Star.

The mail hubs accept mail for hpl.hp.com, so users inside or outside don't need to know what system someone gets their mail on to send to them, as long as they know the registered username or firstname_lastname for that user.

Workstations at HP Labs are not accessible directly from the Internet outside HP. We arrange for mail addressed to user@system.hpl.hp.com to get to that system via Mail Exchanger (MX) records that point to a set of gateway systems.

We also use MX records to direct mail for down systems to a mail server for queuing. From there, we can manually redirect it if necessary. This means that users don't have to worry that mail will bounce if they turn their workstations off for a weekend or their system crashes. If a workstation will be down for a while, we can change aliases to point to a new system (a coworker's system, say, or a timeshare system). This won't catch mail sent directly to user@host, but we have scripts we can use on the server on which mail is queuing to redirect it to the new host (or new user@host).

The big picture of MX records for each host looks like the list in Figure 1. MX records are tried in order of preference, beginning with the lowest; we set the first one to deliver mail to the system itself. If this fails, due to the sending host being on the other side of a firewall or the system being down, the next MX will be tried.

The second MX record points to our internal mail hub, where mail will be queued if the destination system is down. If the sending system isn't an internal HP system, connecting to this mail hub also will fail. If this happens, or the mail hub is down, the next MX record will be tried.

The next record points to our main external mail gateway. It will accept the mail and deliver it internally. The only reason to try the next MX record should be if this mail gateway is down.

```
hplmango.hpl.hp.com   preference = 10, mail exchanger = hplmango.hpl.hp.com
hplmango.hpl.hp.com   preference = 20, mail exchanger = hplms2.hpl.hp.com
hplmango.hpl.hp.com   preference = 25, mail exchanger = hplms26.hpl.hp.com
hplmango.hpl.hp.com   preference = 30, mail exchanger = hplabs.hpl.hp.com
hplmango.hpl.hp.com   preference = 35, mail exchanger = hplb.hpl.hp.com
```

Figure 1: MX Records for a system (hplmango)

The next MX record points to a backup gateway in another building.

The last record points to a backup gateway in another country.

Although we suggest each user advertise his address as name@hpl.hp.com, and we accept mail for this form of addressing, we can't implement site hiding. (Whether we would want to, if we could, is another story.) As pointed out before, users may register one alias with us and have a local alias pointing to their real username (either because in the past we wouldn't give them the alias they wanted, or because the alias they wanted already belongs to someone else). Customers can (and do) set up mailing lists and other accounts on their workstations, so, return addresses would not work if we took out the hostname portion of the address.

Networking

An area that occasionally has problems due to the lack of central control is that of adding a new workstation to the network. Under a central system, we would know when a new system was to be connected to the net, and could manage that process proactively for both hostname and IP address assignment.

As it is, when a customer gets a new workstation, she is responsible for requesting a new hostname and IP address. The main problem is when a user hasn't planned for the new arrival and wants to get "on the air" immediately. The usual turnaround on new IP addresses is twenty-four hours – even if an address is assigned in minutes, the nameservers won't all know the new address immediately.

Sometimes there are problems when a workstation is "cloned" and an existing hostname is duplicated, because this leads to a duplicate IP address being used. This does not happen very often, though, and is usually easy to detect. It is also fairly easy to remedy. Also, since we still have a lot of ThinLAN, users can add their own network drops. But this also doesn't happen often and any problems that do occur are easy to find and fix.

Security

Security presents something of an issue. We have no ability to force any level of security on the workstations – but then, forced security isn't the best kind anyway.

So, what do we do?

We control the perimeter – We have a firewall between HP's internet and the Internet, and we have centrally administered dial-back modems. Phones are centrally controlled by an unrelated department – you can't get a phone line in without going through the appropriate channels and local modems on workstations are forbidden without appropriate paperwork. We haven't caught anyone with their own unauthorized modem yet.

Most people get their OS from us, so we can fix bugs there before they install it. We also could use Ninstall Star if the problem were big enough.

We try to educate people – usually all at once, around the time of an internal audit. We will be trying to do a better job at this, using channels like our newsletter.

We make tools available. At this point, not many, but there are plenty easily available in HP. We just need to make pointers to them available.

The responsibility for security lies with the owners of the individual systems and their management – just like it does for the security of the papers in their filing cabinets or on their desks. We are available as consultants, and we try to provide as secure a site-wide environment as possible.

Backups?

Users are responsible for their own backups, but some users don't realize the value, or necessity, of them. And even for those who do, there is often something just a little more important to do right now . . .

Most work is done on local disks on standalone workstations. A VERY rough guess is there is about 1 TB of disk space on the Unix systems, of which about half (500 GB) is data.

We provide a script with the OS to do backups to a local tape drive, if the system has one. Many do, some do not. This script can do full or incremental backups of user files. It has minimal local customization that needs to be done (name and type of tape drive, what needs to be backed up). This is no longer good enough.

We are working on better solutions.

We plan to have some sort of subscription network backup service. Customers will have to push data from their workstations, since they probably won't allow us the unrestricted access that complete backups require. We also would like to have a restoration mechanism that will allow customers to easily restore their own files, yet will be secure enough to keep them from getting at anyone else's files.

Users' Alternatives

Of course, there are some people who don't want to manage their own systems – either because of the time it takes, or because it is a subject they are not interested in dealing with.

We do have some alternatives for those customers. They aren't perfect, but they fill some of the bigger gaps.

We have timeshare systems where customers who just want to read and send e-mail and news, do some basic text processing, and so on, can spend most of their time.

Of course, they still need something on their desks – in the far past, this would have been a terminal; more recently it is likely to be a PC or HP workstation.

The workstation might just have a basic OS, with the customer logging on to a central system to do most of his work. He still has to manage the workstation, but there isn't much of a problem managing a workstation that isn't customized and has neither data nor applications on it.

Many of these customers prefer a PC – the applications they want are there, and timeshare access enables them to get any Unix services they want. Of course, there are issues with PC management, but that is beyond the scope of this paper.

If a workstation is going to be used only for windows to log into a shared system, why have a full workstation at all? This brings us to our most recent plans – X terminals. A customer can arrange with us to boot her X terminal off a shared system, so that all she has to do is set it up on her desk. She can run her window manager on the server or locally. We are just beginning to work with this X terminal plan, so we aren't sure what the final form will be.

An option of allowing customers to buy an increased level of support has been discussed before, but died for lack of funding. We may try this again eventually.

However, all of these alternatives apply to a relatively small set of our users. Most of our Unix users demand the flexibility and power of managing their own workstations, and find the associated admin work a small price to pay.

Conclusion

Our set up really does seem to work for us.

There are a couple of reasons for this. One is that we have an infrastructure in place. We've had to work at this, but then we also would have to work at putting an infrastructure in place if we did have control – the constraints would just be different.

Another factor is that most of the users here are highly educated engineers and scientists who are sophisticated computer users and capable of managing their own systems.

Probably the most important reason is that our customers accept their responsibility. For example, there is no question of them blaming us when they lose a file because they didn't do backups, since they don't see us as responsible. They usually do make intelligent choices, and we are here to help them do that, to give suggestions and to make things possible.

Author Information

Laura de Leon, Mike Rodriquez, and Brent Thompson are all members of the RCS Site Platform group at the Palo Alto site of Hewlett-Packard Laboratories. They all have the same paper mail address, which is Hewlett-Packard Company, P.O. Box 10490, Palo Alto, CA 94303-0969.

Laura de Leon is a Systems Administrator and Technology Specialist. She has worked at HP Labs for 3 years, since she graduated from Harvey Mudd College, where she received a BS in Mathematics with a Computer Science option, and picked up system administration experience on the side. She was member of the SAGE interim board. She can be reached at deleon@hpl.hp.com.

Mike Rodriquez is a Senior System Administrator by title, and the primary architect of the HP Labs site printing, DNS, and Usenet configurations in reality. He has been programming and administering various flavors of Unix since 1984. Hobbies include traveling to attend concerts and playing with his Multi-Media PC at home. His e-mail address is rodriquez@hpl.hp.com.

Brent Thompson is also a system administrator and software developer. During the past seven years he has primarily been focused on Unix OS updates and system file sharing, and is the primary architect of most aspects of these at HP Labs. His major hobby is growing rare fruits and chilies. His e-mail address is thompson@hpl.hp.com.