

4135

High Availability: The High Cost of Downtime

A. Perry Sellars Jr.
Lead Operational Assessor
Hewlett Packard Company
545 North Pleasantburg Dr.
Suite 100
Greenville, SC 29607 USA
(803) 241-2057

High Availability: The High Cost of Downtime

“It is the day before year-end processing starts. What a year! The new system upgrade we needed finally occurred. This system has been great. That reminds me, when the year end processing is finished I will be able to remove those old disk drives. They have lasted so long. I am as prepared as I can be for year-end. What can go wrong? Boy, I can’t wait until we can get that last good full back-up.” The phone rings. “Hello, Information Services...the system is what??!! What do you mean hung?...Ldev12?... look at the lights on the front,...red and yellow are on solid?....”

As the reader can see, I have left the introduction unfinished. The ending will depend on what this System Manager did during her system upgrade. The planning done during this process could very well determine how this scenario will play out. During the course of this paper we will discuss the various options that she had available to her that might prevent this from ending in a small disaster. High Availability is an ever growing technology. There are many areas that need to be planned before jumping on the High Availability train. How to differentiate between Reliable and Continuously Available (fault-tolerant) environments. How to handle planned and un-planned downtime. Finding out just how much your downtime is really costing you and can the cost of High Availability solutions be justified. How do I plan this environment? While the products and configurations may be different between HP-UX and MPE/iX, the strategic analysis and planning of this environment are very similar. With that in mind this paper will address both platforms from a high-level point of view.

The affects of *planned* and *un-planned* downtime.

While writing this paper I have been looking back over the years I have spent in the data processing field and how uptime requirements have changed. The first shop I worked in was an HP3000 Series III shop. The system was used in a manufacturing environment. We had a next-day hardware support agreement on the system and its peripherals. While we really couldn’t stand the downtime, if we did go down our expectations were that we would be down at least one day. In our environment this was acceptable. The time frame was during the early eighties. I remember a particular week a little before year-end we were having one of those intermittent problems. I am sure you know how tough those can be. Our system was up and down for about three days. Imagine, three days productivity lost. In today’s environment this would be unacceptable.

Downtime whether planned or unplanned can be very costly to a company. In our environment we performed our backups during the off hours after five. While we had a second and third shift in the plant we really didn’t perform other functions that required our system to be available during these hours. We performed our backups, week ending processing and other maintenance activities during these times. So the cost of our planned downtime was insignificant.

In reviewing your environment you need to consider what planned downtime is costing you. Do you have to take your application away from your users to perform a backup? Can you update your application software as well as your Operating System without affecting your users? I would consider activities such as these to be planned downtime operations.

In my customer environments today things have drastically changed. For instance, I have a particular customer that has determined their cost at one million dollars for each hour of downtime. I have another customer that if they have to shut their system down for over thirty minutes they have to shut their production line down! This cost is a very significant amount. Not only in the time they are actually down but the time it takes to stop and start their line. In these environments material waste can be very expensive considering all material that is in the production line when it is shutdown has to be thrown away.

Another example I like to use is a permanent loss of business. In this example imagine you get off of an airplane and walk up to a rental car agency counter. Before you even get the chance to say hello the

clerk tells you their computer system is down and they cannot write up a rental agreement and they aren't sure how long it will be before they can. Your first inclination would be to walk approximately thirty to forty feet to the next counter and rent your car from another company. While the amount of your rental might not appear to be significant, imagine that occurring all over the country at the same time!! This business is permanently lost! This can add up to a very significant loss of revenue.

I can imagine you the reader saying to yourself, but, I'm not in the rental car business. You still need to ask yourself, just how much does my downtime cost me? Lets look at one more example of the cost of downtime.

Recently I was working with a customer in planning their High Availability strategies. Considering the type of business they were in we determined that they could perform their maintenance activities after the stock exchanges closed. They had automated the backup process and could change their tapes during the day. So their planned downtime was really not going to cost them any productivity loss. The question then became do they really need any of the high availability strategies available today? During the planning process we determined that planned or unplanned downtime during the day could very well cost them two hundred thousand dollars a half-hour.

Another area that often goes unnoticed when deciding whether to implement a High Availability solution is the user confidence area. I have a customer that had a very heavily used printer. As a matter of fact I would submit that this printer was one of the most important peripherals on their system. Since the printer was so heavily used it required additional maintenance. These activities required the printer to be down while they were being performed. The information services personnel understood this. But the users did not. As a matter of fact, as far as the users were concerned the printer was "broken" again!! At first this used to bother me. The printer wasn't "broken", it was down so we could perform preventative maintenance to keep it from breaking at an inopportune time. But after looking at this from the users point of view the printer WAS broken. If they couldn't get their printouts when they needed them, the printer was useless to them. Needless to say it did not take too long for the user community to completely lose faith in this printer.

Too much downtime whether planned or unplanned can cause irreparable damage if the user community loses faith in the system. If like the company I used to work for you do not have a need for your system during the off hours then planned downtime can be handled very easily. But that is only if you never have an unplanned downtime event. On the other hand if you are in an environment that can not stand any downtime then products that allow you to perform on-line backup, on-line database management and control are necessities. Again, like one of the previous examples you need to also look at the cost of unplanned downtime.

While today's hardware is more and more reliable it still can and does fail. Lets look at some of the available products today that can help reduce the chances of these failures and also assist in the recovery from ones that do occur.

High Availability Strategies

The first thing we must determine is what we mean by High Availability. I realize this may seem like a moot point but it really isn't. There are many levels of customer uptime needs and therefore there are many different environments to support these needs. The base configuration to start with is the "**reliable**" system. This environment is enhanced by the "**protected**" system products being added. The next step up the availability chain is achieved by adding the "**highly**" available products. The ultimate goal to be achieved is the "**continuously**" available system or better known as fault tolerant system.

A **reliable** system consists of an environment that is robust without spending extra money specifically on High Availability solutions. Items such as an UPS, HP 3000 and 9000 cpu's, disks and tapes. A very well tested operating system as well as the support services to insure the hardware and software are maintained in an up to date fashion. These environments are "reliable" without add-on products.

A “**protected**” system is accomplished through recovery time being enhanced by utilizing component redundancy. For example using products such as Mirror Disk/ux, Mirror Disk/iX, disk arrays, Journaled File System will greatly enhance the recovery time if a failure were to occur. In the protected environment the goal is automatic detection and recovery from faults through redundant hardware and the software needed to support it. One thing to keep in mind, these different systems build one on top of the other to ensure there is a solid foundation to allow the next level to be successful.

To the reliable and the protected system we add the products necessary to achieve the “**highly available**” system. In this environment through the use of products such as SharePlex/iX-Netbase and MCSserviceGuard, the goal is to recover from failure with a very short **application** interruption. Fast recovery with minimal impact. In order to reach this goal this system must have redundant system components and software support which allows an application to be transferred to another system. The time frames for this to take place varies from one to thirty minutes. Also by using the Journaled File System even the one minute time frame can be reduced!

The top rung on the ladder is the “**continuously available**” or fault tolerant system. A continuously available environment has transparent failure recovery, twenty four hour availability, no planned downtime. While this sounds very close to highly available the difference is in this environment the goal is to never have to recover. It is not necessary for instance to justify the need to power down the systems for a repair or update of any kind. While this goal is achievable, it is not without significant cost. Not only in the systems and software requirements but the physical plant requirements as well.

The **reliable** system.

Today hardware has reached a very high mark in reliability. Mean Time Between Failure has increased dramatically over the last few years. While we use formulas to calculate Mean Time Between Failure as well as other measures such as MTTR-Mean Time To Repair, AFT-Annualized Failure Rate, I feel that you already know that the hardware has improved drastically. It is not difficult today to have a reliable environment. There are some things that can be done to enhance this environment without adding additional cost.

In an HP3000 environment by utilizing User Volume Sets you can very well improve your recovery time in case of a failure. By using the standard System_Volume_Set and applying all of your applications in this environment, when you encounter a failure of any of the disk you have to recover the entire system. By dividing your disk sub-system up using User_Volume_Sets you may only have to recover the volume that was affected by the failure. Lets say for instance you have a failure on the system disk. After the disk was repaired you would effectively “install” the system back on that volume set and re-configure your user volume sets. You would not have to “restore” the rest of your system. Not only can this save recovery time but also save time in not having to rebuild your databases or repair other application type problems that occur as a result of a failure.

It does take time to plan how you want to lay out your User Volume Sets but I would contend that it is well worth the time to plan and implement this strategy. Backup time can be reduced as well if you utilize User Volume Sets. The System_Volume_Set would not require backing up as often as the user volumes do. This not only saves time but tapes as well. Using user volume sets can also assist you in planning your backup strategies. If you have a particular user group that the only time they are not on the system is during lunch, then use that time to back that volume set up. This way they have their system when they need it. A main point that I would like to bring out regarding User Volume Sets is the cost. It is FREE!

By keeping your hardware and software up to date and maintaining support agreements in case you do have a failure as well as utilizing a UPS, achieving a “reliable” system is not very difficult.

The *Protected* system

In the protected environment the goal is automatic detection and recovery from faults through redundant hardware and the software needed to support it. You will have to purchase the hardware and software necessary to put this environment in place.

In an HP9000 environment by utilizing Mirror/UX software and duplicating your disk sub-system you can drastically reduce the effects of a disk failure. As an example let's look at a system that has the need for four disk drives.

The first is the root and primary swap volume the next three house the database, application and user space. In the Mirror/ux environment you do not have to duplicate all the hardware. If for instance you have the space available and you only want to mirror the root and swap volume, you can do this on any of the other volumes. One common mistake users make in setting up their root mirror is they don't mirror all the necessary file systems to insure that the system will remain operative in case you lose the root volume. In a 9.0 system for instance you need to insure you have mirrored the usr file system as well the entire root directory, i.e. /dev, /etc and /bin as well as primary swap to insure your system will remain operative if the primary disk fails.

Another area that can be overlooked is the need to have the "mirror" disk on a different I/O channel. To fully take advantage of what you are trying to accomplish with mirroring you need to have your "secondary" or "mirror" volume on a completely different I/O path. While this is NOT a requirement, I would like to suggest that without doing this you really are only protected in case of a disk failure. The main point in mirroring in my opinion is to keep the system from going down in case you have a failure in the disk sub-system. This is to prevent a single point of failure. This includes cabling and termination of the I/O bus.

Taking our example a little further. Let's say we can put our root and primary swap on one disk. We then need to mirror that entire disk to another disk on a separate I/O channel. The other three disk should also be mirrored on this second I/O channel. Not only will the secondary disk be available to keep the system running in case of a disk failure you now are covered if you lose the I/O channel that the disks are attached to. While you might not feel you need to mirror the user space if you lose that disk you are still in a recovery mode that will impede your users.

By utilizing the Mirror/iX product in an HP3000 environment you can also be protected. In this environment the mirror or "secondary" disk has to be of the same type as the primary or on-line disk. As of this article the System_Volume_Set master can not be mirrored. In order to insure a the system disk is in a high availability state, I would suggest that you utilize the User Volume Set approach and install your System_Volume_Set on an array disk.

The User Volumes can be on stand-alone disks that are mirrored and with the system on an array you are covered in case of failure. In both environments the acronym that is used to describe a stand alone configuration is JBOD or Just a Bunch Of Disk. One of the advantages to the JBOD configuration is the flexibility in placement of data, power source and racking.

Mirror/ux as well as Mirror/iX handles the access to the multiple copies of the data whether you are in a failure or a normal mode of operation. By utilizing mirror disks you would be implementing RAID level 1. RAID is an acronym for Redundant Arrays of Inexpensive Disks.

There are several different RAID levels. Level 0 has no check disk, no data protection and is sector interleaved or block striped across a group of disks. Level 0/1 is sector interleaved groups of mirrored disks. Level 1 is mirrored disks. Level 2 is multiple check disks using Hamming Code. Level 3 uses a single check disk using Parity and is byte interleaved or byte striped. Level 4 is a single check disk using Parity, sector interleaved. Level 5 has no single check disk, data and parity spread across all disks, sector interleaved or block striped. Since there is no data protection with RAID level 0 the only benefit is the potential for increased performance due to the data being spread across multiple disks.

To make this environment even more highly available I would suggest that you utilize Disk Arrays in a mirrored configuration. A disk array can prevent data loss by utilizing a “parity” disk. If one of the mechs fail, the parity mech will keep the disk operational until the failed mech can be replaced. This replacement can take place on-line without having to power the disk drive down. After the failed disk has been replaced a re-build function is then performed on-line and the data can be re-built by using the parity mech. By using both technologies you are eliminating many different types of failures.

The one area that still can be a problem is data corruption. If you have data corruption occur on one of the primary disk, then that same corruption will be copied to the mirror disk. There is still the need for performing a backup.

If you are going to implement mirroring as well as disk arrays, you need to insure that you have the underlying items such as power, air conditioning and other environmental needs well planned. What good is mirroring or disk arrays if your power conditioning is too poor to prevent data corruption due to brown outs or other power related problems?

While we have spent quite a bit of time discussing disk sub-systems there are other areas that require attention as well. If your environment is very much dependent upon its LAN or WAN, then you need to take this into consideration.

I had a customer once who utilized our SNA products to communicate with their corporate mainframe. In a lot of cases customers generally only have one link to the mainframe. This customer however had strict uptime requirements. Not only did they have a leased-line access to the mainframe. They also had an X.25 link as well as a satellite link as backup in case the leased-line went down. Not only were these links in place at all times they were tested on a regular basis to insure they would work if they were needed.

In the Local Area Network environment hardware such as bridges, hubs and routers should be duplicated or at the bare minimum alternate routes set up in case of failure. In many cases today the failure of the LAN is at least equivalent to the system failing. Don't forget the goal of a protected system is automatic detection and recovery from faults through redundant hardware and the software needed to support it.

The *Highly Available* system

While the goal in the protected system environment is recovery time being enhanced, in the highly available environment the goal is to reduce the amount of time the application is down while recovering. Let's go over that one more time. For instance, in the protected environment a system should not shut down due to a disk failure. But what happens if your CPU dies? While our data may be protected due to disk redundancy our application is down until the CPU is repaired. In the highly available environment there will be a slight interruption to the application while it is being “switched” to another CPU to continue running. In the HP9000 environment the product that can be used for this is MCSserviceGuard. In the HP3000 environment we utilize SharePlex/iX-NetBase.

MCSserviceGuard - Conceptual Overview

While some of the readers may be familiar with SwitchOver/UX, MCSserviceGuard is a dramatically different product. I would like to take a little time to go over SwitchOver/UX.

The switchover environment was designed to offer a standby solution in case of a CPU failure. In this environment the two CPU's had to be the same hardware type. Disc sub-systems while being cabled together were not shared. The primary CPU transmitted a constant heartbeat and the standby listened. If the primary ceased to send the heartbeat the standby CPU would shutdown, assume the root disk of the primary, reboot and then continue to run the applications that the primary had been running. Here

we see one of the first limitations of SwitchOver/UX. If the standby CPU had been running a different application it would stop running until the primary CPU was back in operation. While ensuring the primary application continued was the main objective you would temporarily lose the standby's application.

Given some of the downtime examples we looked at in a previous section this approach would still be acceptable to sustaining the loss due to a down CPU. One positive of the SwitchOver/UX environment was the protection against a single point of failure. This should be the top priority in any highly available environment. You could have one standby CPU for up to 3 primaries in a SCSI or 7 primaries in a HP-FL disk environment. On the limitation side of the fence SwitchOver/UX does not have the ability to detect an application or service failure. As well there isn't the option to have a standby LAN card that could switch automatically in case the primary card failed. Time to reboot after a failure of course was dependent on how long it took to FSCK the root volume as well as the normal time that a reboot would take.

While there are many positives to the SwitchOver/UX product, I feel that MCSERVICEGUARD offers much more flexibility. In the current release of MCSERVICEGUARD you can have from one to four CPU's in the cluster. These CPU's do not have to be the same hardware type. An E55 can serve to standby for an application running on a T500. While I wouldn't suggest that big of a difference, it is supported. In this environment you are concerned more with the application being available than if we have a CPU failure.

While SwitchOver/UX monitored the CPU only and if a failure occurred there was a complete fail-over switch done, MCSERVICEGUARD monitors not only the CPU but applications, services and other items that each member of the cluster depend on to run. MCSERVICEGUARD allows all of the nodes in the cluster to interrogate each other checking to see if the application or service being monitored is in fact operating. Through the use of multiple LAN interfaces not only can you guard against a LAN card failure but also failures that are outside of the card. If a failure occurs the LAN interface can be switched automatically with only a slight interruption of LAN traffic.

Through clustering, the nodes cooperate to increase the availability of a service. In case of an application failure it can resume operations in a very short period of time. Only one cluster node will run a package at any given time. The package manager ensures this. Node failures such as panics, powerfailures, hangs and network card failures or disconnects (whether un-terminated or other transmission medium problems) along with abnormal termination of a monitored process are just a few of the failure types that MCSERVICEGUARD will monitor for.

Unlike the switchover environment, MCSERVICEGUARD does not "take over" the failed CPU's root disk. This means we don't use valuable time performing an FSCK on a root volume before we can re-start the application. A typical failure and recovery would follow this sequence. Node "cba" detects a cluster application on Node "abc" has failed. The application or "package" is then switched to Node "cba". Before the package is started, Node "cba" does the following. A "start" script is ran that among other things activates the volume group that the application needs in order to run. It also "assumes" the Network Address so the application can still be reached. Runs a crash recovery routine on the package and then starts the application. WHEW! And the time between failure detection and executing the "run" script on the new node occurs in less than a minute (assuming JFS is being used).

As I said earlier, MCSERVICEGUARD is a very flexible product. One thing to keep in mind when planning this environment is the possible need to obtain some consulting during both the planning and implementation phases. The main goal of MCSERVICEGUARD is to reduce the amount of interruption time to an application due to a failure.

SharePlex/iX-NetBase - The highly available solution for the HP3000

To ensure greater system uptime in your HP3000 environment, I would like to suggest SharePlex/iX-NetBase. This product, much like MCSERVICEGUARD is very flexible. With SharePlex/iX-NetBase you

can loosely couple or cluster your systems. This provides you with much greater flexibility by the environment providing a single-system view to the user, cluster operations and management features, shared, cluster-wide facilities for print queues, batch queues, file systems and peripherals and network topology options. SharePlex/iX-NetBase can span over Local as well as Wide Area Networks.

The main objective of SharePlex/iX-NetBase is to provide application and system availability. Since SharePlex is supported over a WAN you also have the added benefit of having your own “disaster” recovery system in case of a “geographical” disaster.

Another feature to keep in mind is the opportunity to move an application from one system in the cluster to another for performance reasons. In the SharePlex environment you can have different types and sizes of CPU's. MPE as well as MPE/iX.

Another major feature of SharePlex/iX-NetBase is the option to use peripherals on the other systems in the cluster. Printers for instance. SharePlex ensures the spoolfile that has been “moved” to a different system printer is actually printed before the original copy is destroyed.

Just as in the MCServicGuard environment you must work to eliminate all single points of failure. Redundant disk's, multiple paths to the disk's, additional LAN cards for protection, mirroring software along with disk arrays for full disk protection.

By utilizing the HPOpenView System Manager the operator can have one “view” of the systems they are supporting. HPOpenView System Manager is a very straight-forward product that can be utilized right away. Many of the operator functions can be automated, tested and verified complete using HPOpenView System Manager. The minimum requirements are at least two HP3000's networked together with SharePlex/iX running on both systems.

SharePlex/iX-NetBase is a product that can come “bundled” with many different options. Master Print/Spooling Management - to allow a user to print to any spooled printer on the network as long as it is configured in SharePlex/iX, Network File Access - which allows users and applications access to data and programs on the other HP3000's in the cluster, Shadowing - which provides a complete system and data/application replication system. The Shadowing is an entry level product. SharePlex/iX can be order as the “bundle” to include all, or as entry level which would include the shadowing product only.

Other things to consider with SharePlex/iX is the opportunity to move users and applications to a different system in the cluster while performing maintenance activities on their “home” system. Just like MCServicGuard I would suggest that SharePlex/iX requires a fair amount of planning. Not only to install and configure but the “what ifs” as well. The flexibility of SharePlex/iX-NetBase coupled with the additional products of Print Management and NFA make it a most viable product to ensure your HP3000 is in a highly available environment.

The most important thing to remember is to not have any single points of failure. If your disk sub-systems aren't protected by either mirroring, disk arrays or both, what good will MCServicGuard or SharePlex/iX-NetBase do for a disk failure? Using the same logic, while your disk systems may be protected through these measures what happens if you have a failure on the CPU or LAN? For MCServicGuard or SharePlex/iX-NetBase to work to it's maximum level there should not be any single points of failure, duplicate paths to disks, duplicate LAN interfaces, mirrored and high available disk arrays.

The *Continuously Available* system

In the Continuously Available system environment NO downtime is the goal. While this can be accomplished it is not without cost. In many ways this cost is not in the hardware and software alone, but also includes the physical plant as well.

In this environment nothing can be allowed to bring the system down including geographical, natural and other types of disasters. ALL hardware must be redundant. System components as well as peripherals. Not only do they have to be redundant but they must be “hot swap-able”. But more importantly, they have to be able to “switch” to the alternate hardware automatically. If you lose a power supply you should never see a problem. After supplying a warning message and possibly a test-fail led, the system should simply run on the “alternate” supply. The defective supply can then be “hot swapped”.

Hewlett Packard offers a fault-tolerant system, the SPP1200/CD Super Parallel Server. Not only is this system truly fault-tolerant it is very scaleable. Like the previous environments there should be no single points of failure. For instance, in the computer room environment, there has to be back-up power such as a Motor Generator, additional air conditioning that can be automatically switched in case of primary unit failure. Also, the operating system and applications in this environment must be able to be patched and tested without having to shut the system down.

Generally these environments also have very stringent security as well as “hardened” equipment rooms. Don’t forget that some failures can occur as a result of internal sabotage. For more information regarding the HP SPP1200/CD Super Parallel Server and other requirements to ensure a continuously available environment you can call the Hewlett Packard Custom Products Group at (508) 436-4999. Providing the Continuously Available environment requires extensive planning.

Other Considerations

Earlier we discussed planned and un-planned downtime. One issue to keep in mind is the amount of downtime used to perform backups. In the HP3000 environment the standard backup utilizes the **STORE** command. Using this command requires the system be “quiet”. The system is then “down” to the user. There are many ways to prevent this downtime from impacting your users. For the one-shift operation, you can always backup during the off hours. For the twenty-four hour shop though, there are no off-hours. You then must look at other options. While there are many products on the market to allow on-line back-up operations, the TurboSTORE/iX product offered by Hewlett Packard provides many options to assist in this requirement. Please use the comparison chart below to see what product better suites your back-up needs.

FEATURE	Release 5.0						Release 5.5		
	STORE/iX		TurboSTORE/iX				STORE/iX		
	FOS	30319A	30387A	30388A	30397A	30398A	FOS	B5151AA	B5152AA
Multiple Store Devices	-	X	X	X	X	X	X	X	X
File Interleaving	-	X	X	X	X	X	X	X	X
Data Compression	-	-	X	X	X	X	-	X	X
Parallel Restore	-	X	X	X	X	X	-	X	X
On-line Backup	-	-	-	-	-	X	-	-	X
7x24 True-On-line Backup	-	-	-	-	-	-	-	-	X
Store to Disk	-	-	-	-	-	-	-	-	X
Optical Device	-	-	-	-	X	X	-	X	X
Labeled Tapes	X	X	X	X	X	X	X	X	X

It is not the intent of this paper to further examine the options of TurboSTORE but as you can see there are many options to reduce the amount of planned down-time during a back-up procedure. Another way is through the use of User_Volume_Sets. This method was discussed earlier.

Like the HP3000 environment, the HP9000 environment has many different options that can be utilized to perform on-line backups. The one that I would like to highlight in this article is the Hewlett Packard product OmniBack II. OmniBackII provides support of the current on-line backup API’s. System availability is maintained at 100%. The backup API’s that are or will be supported are listed below.

SAP R/3 On-line Backup

Oracle’s Parallel Backup/restore Utility TM

Informix
Sybase

OmniBack II has many more features but the intent of this article is in letting you know that OmniBack II as well as TurboStore/iX 24x7 True On-Line backup can provide you with an alternative to having to shut your system or application down to perform that planned downtime event we all have the need for - Backup!

In Summary:

In today's environment downtime, whether planned or unplanned, is becoming more and more unacceptable. By determining your downtime cost and properly planning you can reduce the effects that an untimely downtime event can have on your productivity. You need to first determine what type of environment you are striving to maintain. Do your needs require a **reliable, protected, highly available** or **continuously available** system? In all of the environments other than the reliable system the primary goal should be to eliminate all single points of failure. By utilizing mirroring, array disks, MCServGuard, SharePlex/iX-NetBase and on-line backup products such as TurboStore 24x7 True On-Line backup and OmniBack II, you greatly improve the availability of your system while reducing the recovery time if an un-planned downtime even should occur.

Oh! I almost forgot!! What happened to our System Manager in the introduction?..Let's look back in and see what happened.

“So...Red and Yellow are on solid? Look at the drive labeled Ldev 24...what do you see?...a flashing green light?...okay...what is the system doing now?...it IS!!..GREAT...that is what was supposed to happen...remember, we installed and configured Mirror/iX when we did the upgrade. Okay, let's be on the safe side..let's get all the users off, get our Full Backup started, and then call the 800 number and get Ldev 12 repaired. I sure am glad we looked into the “protected” system information while planning this upgrade”