

# SESSION 8 – SYSTEM BACKUPS AND DISASTER RECOVERY

**Course Writer: Dr. Ed. Danso Ansong**, Dept of Computer Sc.  
Contact Information: [edansong@ug.edu.gh](mailto:edansong@ug.edu.gh)



## UNIVERSITY OF GHANA

College of Education

**School of Continuing and Distance Education**

- In this session, students shall learn about the importance of backups in disaster recovery.

The key topics to be covered in the session are as follows:

- Backup and Recovery
- Issues associated with Backups
- Backup strategies and Scheduling
- Backup devices and media
- Types of Backup Software

- Refer to the following reading material which is available on Sakai

## **RECOMMENDED TEXT**

- Essential of Systems Administration 3<sup>rd</sup> Edition  
[Chapter 7]

# Chapter Objectives

At the end of the session, the student will be able to:

- Understand importance of backups
- Know the issues associated with backups
- Identify the various backup scheduling strategies
- Understand basics of backup media and devices
- Explore the various types of backup software

.



# Importance of System Backups

- Which Files Should Be Backed Up?
  - OS Binaries?
  - Applications?
  - Configuration Files?
  - User files?
  - Log files?
    - Generally, full backups of everything are easiest to manage, but backup of system files is creating extra work for yourself.
      - **Possibly full dump when installed, then again after patches/upgrades.**
    - Backup of just user files is not enough.
    - Should dump the log files, and configuration information.



# Importance of System Backups

- How Often Should Backups Be Performed?
  - Need to determine what level of data loss is acceptable:
    - Web sales? - need very fine grain backups.
    - Banking/Insurance? - very fine grain.
    - Research and development? - fine to medium grain.
    - University? - medium grain.
    - Mom and Pop? - coarse grain.



# Backup Strategy and Scheduling

- Types of Backups
  - Full backup– dump every file to dump media.
  - Partial (incremental) backup – dump all files that have changed since last lower level backup to dump media.
    - Unix uses a multi-level partial scheme (level 0 is a full, level 1 – 9 are incremental).
      - All files that have been modified since the last ufsdump at a lower dump level are copied to the dump media.
      - For instance, if a "level 2" dump was done on Monday, followed by a "level 4" dump on Tuesday, a subsequent "level 3 dump on Wednesday would contain all files modified or added since the "level 2" (Monday) backup.
      - A "level 0" dump copies the entire file system to the dump media.





# Backup Strategy and Scheduling

- Backup Strategies
  - There are several algorithms that might be used to schedule full and partial backups.
  - The choice of algorithm dictates the amount of media required.
  - The choice of algorithm plays a large role in the size of the restore window (how long is data available from a backup tape).
  - Some of the more popular algorithms are:
    - Volume/Calendar Backup
    - Grandfather/Father/Son Backup
    - Tower of Hanoi Backup



# Backup Strategy and Scheduling

- Volume/Calendar Backup
  - The volume/calendar backup strategy calls for a full system backup once a month.
  - An incremental backup is performed once a week for files that change often.
  - Daily incremental backups catch files that have changed since the last daily backup.
  - A typical schedule would be to perform the full (level 0) backup one Sunday a month, and weekly level 3 backups every Sunday of the month.
  - Daily level 5 backups would be performed Monday through Saturday.
  - This would require eight complete sets of media (one monthly tape, one weekly tape, and six daily tapes)



# Volume Calendar

	Su	M	Tu	W	Th	F	Sa
<b>Week 1 Tape</b>	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>		<b>E F</b>	<b>G</b>
<b>Level</b>	<b>0</b>	<b>5</b>	<b>5</b>	<b>5</b>	<b>5</b>	<b>5</b>	<b>5</b>
<b>Weeks 2, 3, 4</b>	<b>H</b>						
<b>Level</b>	<b>3</b>	<b>5</b>	<b>5</b>	<b>5</b>		<b>55</b>	<b>5</b>

- Recovering from complete data loss with the volume/calendar scheme requires restoring from the most recent full backup, then restoring from the most recent weekly backup, and finally, restoring from each daily backup tape written since the weekly backup.
- An advantage to this backup scheme is that it requires a minimum of media, but this also points out one of the major problems with this backup scheme: tapes are immediately reused.
  - For example, every Monday overwrites last Monday's backup information. Consider what would happen if one of the disk drives failed during the second Monday backup.
  - It would not be possible to recover all data, because the system was in the process of overwriting the backup tape when the drive failed.



# Backup Strategy and Scheduling

- Grandfather/Father/Son Backup
  - The grandfather/father/son backup strategy is similar to the volume/calendar strategy.
    - The major difference between the two schemes is that the grandfather/father/son method incorporates a one-month archive in the backup scheme. This eliminates the problem of overwriting a tape before completing a more recent backup of the file system.
    - Implementing the grandfather/father/son strategy requires performing a full (level 0) dump once a month to new media.
    - Once a week, an incremental (level 3) backup must be performed that captures all files changed since the last weekly backup.
    - This weekly backup should also be saved on new media.
    - Each day an incremental level 5 backup must be performed to capture files that have changed since the last daily backup.
    - The daily backups reuse the tapes written one week earlier



# Grandfather/Father/Son

	Su	M	Tu	W	Th	F	Sa	
Week 1 Tape A			B	C	D	E	F	G
Level	0		5	5		5	5	3
Week 2 Tape H								I
Level	5							3
Week 3 Tape H								J
Level	5							3
Week 4 Tape H								K
Level	5							3
Week 5	L							M
Level	0							3
Week 6 Tape H								G
Level	5							3
Week 7 Tape			H					I
Level	5							3
Week 8 Tape H								J
Level	5							3
Week 9 Tape A								
Level	0							



# Grandfather/father/son

- To maintain a one-month archive, the monthly full backup tape should be placed in storage.
- Each weekly full backup should be placed in storage.
- The second monthly full backup, should use new media.
- When the third monthly backup is due, the first month's full backup media should be reused. The weekly backups are archived in a similar manner.
- This scheme requires two sets of monthly backup media, five sets of weekly backup media, and six sets of daily backup media.
- A total of 13 sets of media are required to implement this strategy with a one-month archive of information.
- To recover from complete data loss, first restore the most recent level 0 backup tape.
- Next, restore from the most recent of the level 3 backups, if that backup was written after the level 0 backup.
- When the level 3 backup has been restored, the operator would restore from each of the level 5 backups written after the level 3 backup.
- This backup strategy requires much more media than the simple volume / calendar strategy. Although media cost is increased with this plan, data survivability also increases.



# Backup Strategy and Scheduling

- Tower of Hanoi Backup
  - The Tower of Hanoi backup strategy is a variation of “exponential backup.” Both strategies rely on functions of powers of 2.
  - For example, the use of five backup tapes provides for a 32-day schedule. The use of six tapes would provide for a 64-day schedule.
  - The Tower of Hanoi backup schedule provides outstanding data survivability and a minimum of media. Unfortunately, on a seven-day backup system, the scheduling of full backups as opposed to partial backups can become a problem for the operator.
  - One way to avoid operator confusion is to perform a special level 0 backup on the first day of each month. This tape would not be one of the five tapes used in the backup cycle. Total media requirements in this scheme would be seven sets of media.



	Su	M	Tu	W	Th	F	Sa				
Week 1 Tape	E	A	B	A	C	A	B				
Level	0	5	4	5		3	5	4			
Week 2 Tape	A	D		A	B		A		C	A	
Level	5	1		5		4	5		3	5	
Week 3 Tape	B	A	E		A		B		A	C	
Level	4	5		0		5	4		5	3	
Week 4 Tape	A	B	A		D		A		B	A	
Level	5		4	5		1	5		4	5	
Week 5		C	A	B		A	E		A	B	
Level	3	5		4		5	0		5	4	
Week 6 Tape	A	C	A		B		A		D	A	
Level	5	3		5		4	5		1	5	
Week 7 Tape	B		A	C		A		B		A	E
Level	4	5		3		5		4		5	0
Week 8 Tape	A	B		A	C		A		B	A	
Level	5	4		5		3		5		4	5
Week 9 Tape	D	A		B							
Level	1	5		4							



# Backup Strategy and Scheduling

- Tower of Hanoi Backup
  - To recover from complete data loss, first restore from the most recent level 0 backup, and then restore from the level 1 backup if that backup was written after the level 0 backup.
  - Next, restore consecutively from the most recent level 3 and 4 backups if both were written after the level 0 backup.
  - Finally, restore each of the level 5 backups that were written after the level 0 backup.



# Reasonable Alternative

- The following four-week schedule offers a reasonable backup schedule for most sites.
- Performing a full dump on the first Sunday of the month provides a monthly snapshot of the system data.
- Using two sets of dump media allows the operator to store information for two full months.
- Note that in the example the Tuesday through Friday incremental backups contain extra copies of files from Monday.
- This schedule ensures that any file modified during the week can be recovered from the previous day's incremental dump.
- To recover from complete data loss, restore the most recent full (level 0) backup tape.
- Next, restore from the most recent of the weekly (level 3) backups. Once the weekly backups are restored, restore from each of the daily (level 5) backups.



# A Reasonable Alternative?

	Su	M	Tu	W	Th	F	Sa	
<b>Week 1 Tape</b>			A	B	C	D		E
<b>Level</b>			0	5	5	5	3	
<b>Week 2 Tape</b>			F		B		C	D
<b>Level</b>			5		5	5	5	3
<b>Week 3 Tape</b>			F		B		C	D
<b>Level</b>			5		5		5	3
<b>Week 4 Tape</b>				F		B	C	D
<b>Level</b>				5		5	5	3



# Backup Devices

- Backup devices must exhibit the following traits:
  - User ability to write data to the device.
  - Media capable of storing the data for long periods.
  - Support of standard system interconnects.
  - Support of reasonable input/output throughput.



# Backup Devices

- Tape Backup Devices
  - Cartridge Tape Drive
  - 8-mm Tape Drive
  - Digital Audio Tape Drive
  - Linear Tape Open
  - Digital Linear Tape
  - Jukebox/Stacker Systems
- Optical Backup Devices
- Magneto-optical Backup Devices
- Disk Systems As Backup Devices
  - RAID Disk Arrays
  - Problems with Disks As Backup Devices
- High-Density Removable Media Backups



# Backup Devices

- Tape backup devices are probably the most common backup media in use.
  - The media is relatively inexpensive, the performance is reasonable, the data formats are standardized, and tape drives are easy to use.
  - These factors combined make magnetic tape backups an attractive option.
  - Most current-generation tape drives offer “native” mode and compressed mode storage capabilities.
    - Generally, the manufacturers claim a 2:1 compression ratio, but this value may vary based on the data to be stored.
    - Binaries (images, compiled programs, audio files, and so on) may not be significantly smaller when compressed, whereas text files may compress very well.



# Tape Backup Devices

- **Cartridge Tape Drive**

- Cartridge tape drives store between 10 Mb and several Gb of data on a small tape cartridge.
- Most cartridge tape systems use SCSI interconnections to the host system.
- These devices support data transfer rates up to 5 Mb per second. The actual transfer rate from the tape drive memory to the tape media is typically about 500 Kb per second.

- **8-mm Tape Drive**

- These tape drives are also small and fast, and use relatively inexpensive tape media.
- The 8-mm media can hold between 2 and 100 GB of data, depending on the drive model and type of tape in use.
- The 8-mm drives use the SCSI bus as the system interconnection.
- Low-density 8-mm drives can store 2.2 Gb of information on tape. and transfer data to the tape at 250 Kb per second. High-density 8-mm drives can store up to 80 GB of information on a tape at a 16 MB/second.
- “low” end, the 8-mm drives do not use data compression to store the data on tape. “high” end, advanced intelligent tape drives incorporate compression hardware and improved recording techniques to increase the amount of information that can be stored on the tape.



# Tape Backup Devices

- **Digital Audio Tape Drive**

- Digital audio tape (DAT) drives are small, fast, and use relatively inexpensive tape media. Typical DAT media can hold between 2 and 40 GB of data.
- Although manufacturers of DAT devices have announced the end-of-life for these products, they will remain in use for many years.
- The various densities available on DAT drives are due to data compression. A standard DAT drive can write 2 Gb of data to a tape. By using various data compression algorithms, and various lengths of tape, manufacturers have produced drives that can store between 2 and 40 GB of data on a tape.
- DAT drives use SCSI bus interconnections to the host system, and typically offer 3 MB/second throughput.





# Tape Backup Devices

- **Linear Tape Open**

- A consortium of Hewlett Packard, IBM, and Seagate developed the LTO technology. LTO encompasses two formats: the Ultrium, a high-capacity solution, and Accelis format, a fast-access format. The two formats use different tape drives, and tape cartridges.
- LTO Ultrium drives can store up to 100 Gb of data on a single tape cartridge at 16 Mb/second.



# Tape Backup Devices

- ***Digital Linear Tape***

- Digital linear tape (DLT) backup devices are also relatively new on the backup market.
- These tape devices offer huge data storage capabilities, high transfer rates, and small (but somewhat costly) media.
- Digital linear tape drives can store up to 110 Gb of data on a single tape cartridge. Transfer rates of 11 Mb/second are possible on high-end Super-DLT drives, making them very attractive at sites with large on-line storage systems.
- Where 8-mm and DAT tapes cost (roughly) \$15 per tape, the LTO, AIT, and DLT tapes can run as much as \$150 each. However, when the tape capacity is factored into the equation, the costs of these high-capacity tapes become much more reasonable.
  - **Consider an 8-mm tape that holds (up to) 14 Gb on average versus a LTO cartridge, which can hold 100 Gb of data!**



# Tape Backup Devices

- *Jukebox/Stacker Systems*

- Jukebox or stacker systems combine an automated mechanism with one or more tape drives.
- Stackers are sequential tape systems. Tapes are stacked in a hopper, and the tape drive starts by loading the tape at the bottom of the stack. When the tape is full, it is ejected, and the next tape is loaded from the stack.
- Many stackers do not have the capability to load a specific tape in the drive. Instead, these stackers simply cycle (sequentially) through the tapes until the last tape is reached. At this point they can either start the cycle over again or wait for a new group of tapes to be loaded into the hopper.



# Tape Backup Devices

- ***Jukebox/Stacker Systems***

- Unlike stackers, jukebox systems employ multiple tape drives, and special “robotic” hardware to load and unload the tapes.
- Jukebox systems require special software to control the robotics. The software keeps track of the content of each tape and builds an index to allow the user to quickly load the correct tape on demand. Each tape is “labeled” with a bar-code decal (or something similar), and the mechanism contains a label reader that keeps track of what tape is in the drive. Many commercially available backup software packages allow the use of jukebox systems to permit backup automation.



# Optical Backup Devices

- **Optical Backup Devices**

- Recently, optical storage devices have become another economical means of backing up mass storage systems.
- Compact disk read-only-memory devices (CD-ROM) are useful for long-term archive of information.
  - **Although the name implies that these are read-only devices, recent technology has made it possible to mass market the devices that create the encoded CD-ROM media.**
  - **These CD-ROM writers (also called CD-recordables) make it possible to consider CD-ROM as a backup device. More recent versions of this technology have produced rewritable CD-ROMs (CD-RW or CDR).**



# Optical Backup Devices

- **Optical Backup Devices**
  - One of the major decisions in choosing a backup device is the ability of the medium to store information for long periods.
    - CD-ROM media offer excellent data survivability.
    - Another advantage to the CD-ROM is the availability of reliable data transportability between systems. This reliability is possible due to the CD-ROM's adherence to industry standardized data formats.
    - Along with these advantages, the CD-ROM offers a few unique disadvantages. The foremost disadvantage to the CD-ROM as a backup device is the setup cost to create a CD. Setting up and creating a CD is a time-intensive operation.
  - Some small sites may decide to back up to CD-ROM rewritable (CDR) media. The CDR format allows the reuse of optical media, thereby reducing the cost of backing up to optical devices.
    - Unfortunately, the CDR is still a low-density solution, providing a mere 650 Mb of storage per disk. The setup and record time for CDR is comparable to CD-ROM media, making CDR less attractive for backups at large sites.



# Magneto-Optical Backup Devices

- **Magneto-optical Backup Devices**

- Optical storage systems and associated media are typically expensive. They are also relatively slow devices. Consequently, optical storage systems are rarely used as backup devices at large sites.
  - In contrast, magnetic tape (or disk) storage systems are inexpensive and fast. Unfortunately, the media is bulky and susceptible to damage and data loss.
  - By combining the two storage systems into a single system, manufacturers have been able to provide fast, inexpensive, and reliable backup systems.
- Many of the magneto-optical systems are hierarchical, meaning that they keep track of how long a file has been in storage since the last modification.
  - Files that are not accessed or modified are often eligible to be stored on the slower optical storage section of the system.
  - Frequently accessed files are maintained on the magnetic storage section of these systems, which allows for faster access to files.
- Most magneto-optical storage systems use standard SCSI bus system interconnections. These systems can typically provide the same (or better) data transfer rates as SCSI tape and disk systems.



# Disk Backup Devices

- **Disk Systems As Backup Devices**

- One problem involved in using tape devices for backups is the (relatively) low data throughput rate.
- If the operator had to back up several gigabytes or terabytes of data daily, it would not take long to realize that tape drives are not the best backup method.
- Although optical backup devices offer high storage capacity, the optical devices are often much slower than tape devices.
- One popular method of backing up large-scale systems is to make backup copies of the data on several disk drives.
  - Disk drives are orders of magnitude faster than tape devices, and therefore offer a solution to one of the backup problems on large-scale systems.
    - However, disk drives are much more expensive than tapes.
    - Disk backups also consume large amounts of system resources.
    - For example, you would need 100 2-Gb disks to back up 100 2-Gb disks. Fortunately, there are software applications and hardware systems available to transparently perform this function.





# Disk Backup Devices

- ***RAID Disk Arrays***

- One operating mode of redundant arrays of inexpensive disks (RAID) enables the system to make mirror image copies of all data on backup disk drives.
- RAID disk arrays also allow data striping for high-speed data access.
- Yet another mode stores the original data, as well as parity information on the RAID disks. If a drive should fail, the parity information may be used to recreate the data from the failed drive.

- ***Problems with Disks As Backup Devices***

- When tape devices are employed as the backup platform, it is a simple matter to keep a copy of the backups off-site.
- When disk drives are employed as a backup media, the process of keeping a copy of the backup media off-site becomes a bit more complicated (not to mention much more expensive).
  - **In the case of a RAID disk array, the primary copy of the data is stored on one disk, and the backup copy of the data is stored on another disk. However, both disks are housed in a single box. This makes the task of moving one drive off-site much more complicated.**



# Disk Backup Devices

- RAID disk arrays have recently been equipped with fiber channel interfaces.
- The fiber channel is a high-speed interconnect that allows devices to be located several kilometers from the computer.
  - By linking RAID disk arrays to systems via optical fibers, it is possible to have an exact copy of the data at a great distance from the primary computing site at all times.
- ***High-Density Removable Media Backups***
  - A relatively recent addition to the backup market is the high-density removable media drive.
  - Examples of these devices include the Iomega ZIP and JAZ drives, and the Imation Superdisk drives.
    - These devices are capable of recording 100 Mb to 2 Gb of data on a removable medium that resembles a floppy diskette.
  - Until recently, UNIX could not make use of these high-density removable media devices.
  - Many of these devices employ a parallel port interface. A few of them offer SCSI interfaces, allowing them to be connected to the external SCSI port on a workstation.



# Unix Backup Commands

- Unix operating environments include a plethora of backup utilities.
  - The [ufs]dump and [ufs]restore utilities are available under most UNIX variants.
    - **The dump application was developed to allow the backup of entire systems one at a time.**
    - **The dump program allows the operator to specify files to be “dumped” (or backed up to tape), and options to use during the dump.**
    - **In addition, dump enables scheduling of different levels of dumps on different days.**
    - **The dump command also allows for dumps that occupy multiple tape reels.**



# UNIX Backup Commands

- `dump [options] [arguments] files_to_dump`
  - NOTE: The dump command requires that the user have read access privileges on the system disks.
    - **0-9:** These numeric values specify the dump level. All files that have been modified since the last dump at a lower dump level are copied to the media.
    - **b:** Signifies the blocking factor to be used. The default is 20 blocks per write for tape densities of 6,250 BPI (bytes per inch) or less. The blocking factor of 64 is used for tapes with 6,250 BPI or greater density. The default blocking factor for cartridge tapes is 126. NOTE: blocking factor is specified in 512-byte blocks.
    - **c:** Signifies that the backup device is a cartridge tape drive. The option sets the density to 1,000 BPI and the blocking factor to 126.
    - **d:** Signifies the density of the backup media in BPIs. The default density is 6,250 BPI except when the c option is used. When the c option is used, the density is set to 1,000 BPI per track.



# Unix Backup Commands

- D: Signifies that the dump device is a floppy diskette.
- f: Signifies the dump file. This option causes dump to use dump file as the file to dump to, instead of /dev/rmt/0.
- s: Signifies the size of the backup volume. This option is not normally required because dump can detect end-of-media. When the specified size is reached, dump waits for the operator to change the volume. The size parameter is interpreted as the length in feet for tapes and cartridges, and as the number of 1,024-byte blocks for diskettes.
- u: This option causes dump to annotate which file systems were dumped, the dump level, and the date in the /etc/dumpdates file.
- v: This letter signifies that dump should verify the content of the backup media after each tape or diskette is written.
- # dump 0fu /dev/rmt/0 /dev/rdisk/c0t3d0s2



```
BACKUP: host dump.plc.com : level 0 : filesystem /
DUMP: Connection to dump.plc.com established.
DUMP: Date of this level 0 dump: Sat Feb 16 00:20:40 2002
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /dev/ida/c0d1p2 (/) to /dev/nst0 on host dump.plc.com
DUMP: Label: none
DUMP: mapping (Pass I) [regular files]
DUMP: mapping (Pass II) [directories]
DUMP: estimated 6094028 tape blocks.
DUMP: Volume 1 started at: Sat Feb 16 00:20:42 2002
DUMP: dumping (Pass III) [directories]
DUMP: dumping (Pass IV) [regular files]
DUMP: 15.47% done, finished in 0:27
DUMP: 29.66% done, finished in 0:23
DUMP: 42.42% done, finished in 0:20
DUMP: 54.50% done, finished in 0:16
DUMP: 68.57% done, finished in 0:11
DUMP: 83.08% done, finished in 0:06
DUMP: 97.86% done, finished in 0:00
DUMP: Closing /dev/nst0
DUMP: Volume 1 completed at: Sat Feb 16 00:58:18 2002
DUMP: Volume 1 took 0:37:36
DUMP: Volume 1 transfer rate: 2846 KB/s
DUMP: 6421153 tape blocks (6270.66MB) on 1 volume(s)
DUMP: finished in 2256 seconds, throughput 2846 KBytes/sec
DUMP: level 0 dump on Sat Feb 16 00:20:40 2002
DUMP: Date of this level 0 dump: Sat Feb 16 00:20:40 2002
DUMP: Date this dump completed: Sat Feb 16 00:58:18 2002
DUMP: Average transfer rate: 2846 KB/s
DUMP: DUMP IS DONE
```

# █

# UNIX Backup Commands

- Now that file systems have been copied onto a tape, how is this information retrieved?
  - UNIX provides an application to restore data from the backup media to the system mass storage devices.
  - This application is called *restore*.
- `/usr/sbin/restore options [ arguments ] [ filename ... ]`
- Some of the most useful options to the *restore* commands follow.
  - ***i***: Places *restore* in the interactive mode. Commands available in this mode follow.
  - ***add [filename]***: Adds the named file or directory to the list of files to be extracted.
  - ***cd directory***: Changes to directory on the dump media.
  - ***delete [filename]***: Deletes the current directory or file from the list of files to be extracted.
  - ***extract***: Extracts all files on the extraction list from the dump media.



# UNIX Backup Commands

- ***ls [directory]***: Lists files in directory (dump media) or the current directory, which is represented by a period (.).
- ***pwd***: Prints the full path name of the current working directory.
- ***quit***: Exits immediately.
- ***verbose***: Toggles the verbose flag (the program prints a line for every action it takes).
- ***r***: Restores the entire content of the media into the current directory.
- ***x***: Extracts the named files from the media.
- ***b***: Sets the restore blocking factor.
- ***f [dump file]***: Tells *restore* to use the *dump file* instead of */dev/rmt/0* as the file to restore from.
- ***R***: Resumes *restore* after volume is changed.
- ***t***: Prints table of contents for *dump* file.
- ***n***: Skips to the *n*th file when multiple *dump* files exist on the same tape.
- ***v***: Displays the name and *inode* number of each file restored.
- Note that the *i*, *r*, *R*, *t*, and *x* arguments are mutually exclusive. Only one of these arguments may be used at a time.

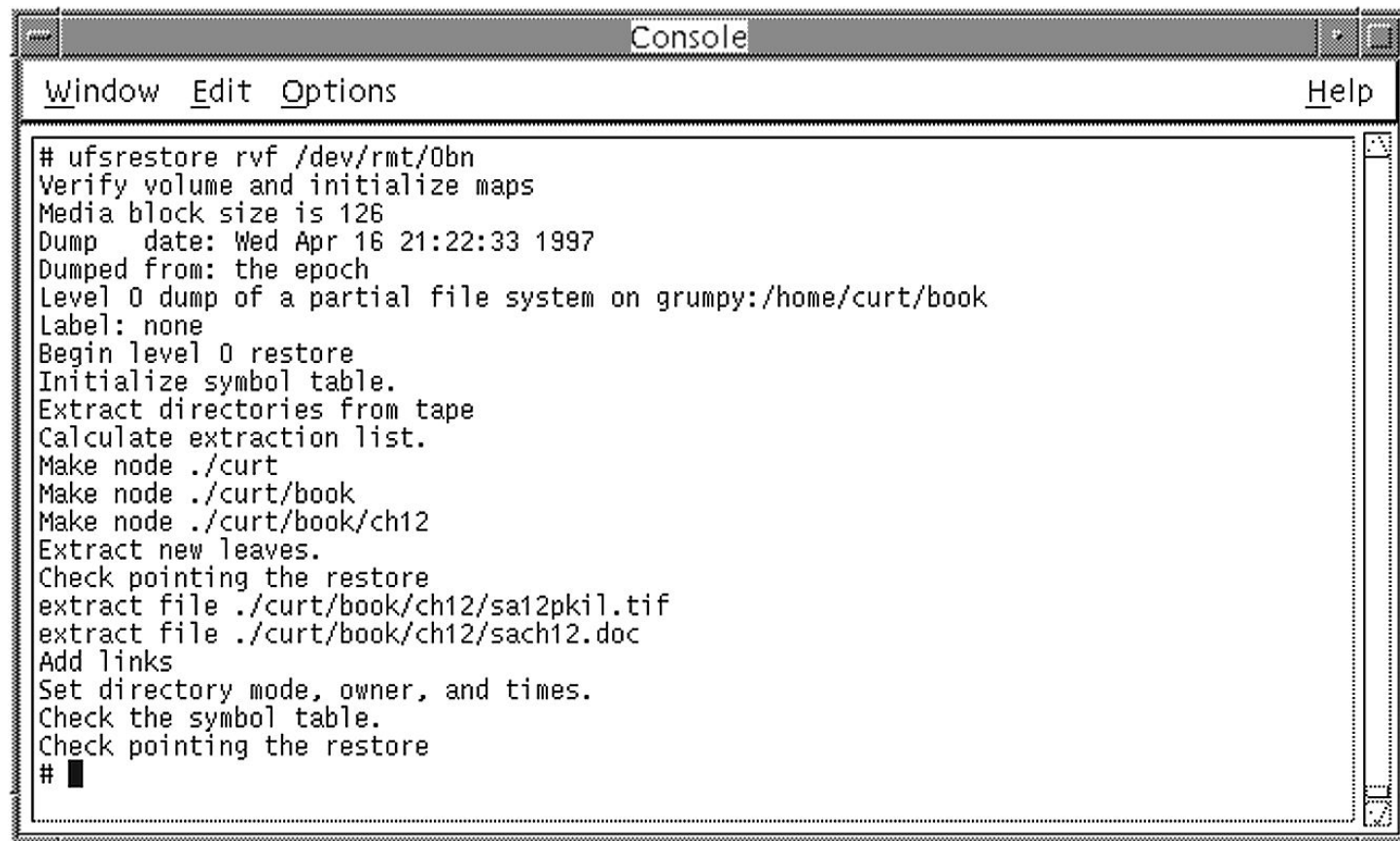




# Unix Backup Commands

- Example restore Commands

```
# restore rbf 50 /dev/rmt/0 /dev/rdisk/c0t2d0s2
```



```
# ufsrestore rvf /dev/rmt/0bn
Verify volume and initialize maps
Media block size is 126
Dump   date: Wed Apr 16 21:22:33 1997
Dumped from: the epoch
Level 0 dump of a partial file system on grumpy:/home/curt/book
Label: none
Begin level 0 restore
Initialize symbol table.
Extract directories from tape
Calculate extraction list.
Make node ./curt
Make node ./curt/book
Make node ./curt/book/ch12
Extract new leaves.
Check pointing the restore
extract file ./curt/book/ch12/sa12pkil.tif
extract file ./curt/book/ch12/sach12.doc
Add links
Set directory mode, owner, and times.
Check the symbol table.
Check pointing the restore
# █
```



# Unix Backup Commands

# restore ibf 50 /dev/rmt/0

```
xterm
# mt -f /dev/rmt/0 rew
#
# ufsrestore if /dev/rmt/0cbn
ufsrestore > ls
.:
  .ssh/      devices/  etc/
ufsrestore > cd etc
ufsrestore > ls
./etc:
  dfs/      dumpdates  mail/      rmtab
ufsrestore > cd mail
ufsrestore > ls
./etc/mail:
  block..db.old  block.db      block.db.old  sendmail.pid  statistics
ufsrestore > add block.db
ufsrestore > extract
You have not read any volumes yet.
Unless you know which volume your file(s) are on you should start
with the last volume and work towards the first.
Specify next volume #: 1
set owner/mode for '.'? [yn] y
ufsrestore > quit
# █
```



# Unix Backup Commands

- Remote Backup and Restore
  - How can a file system dump be performed on a system without a backup device?
  - The *dump* and *restore* commands allow input and output to be sent to the standard input and output streams. This flexibility allows the dump/restore output to be sent through the network to a system with a backup device.
  - The *-f* option for *dump* and *restore* specifies the dump device.
  - *f dump file*: This option tells *dump* to use *dump file* as the file to dump to. If *dump file* is specified as a hyphen ( - ), *dump* will dump to standard output. If the name of the file is of the form *machine:device*, the dump is carried out from the specified machine over the network using the *rmt* facility.
  - *NOTE*: Because *dump* is normally run by *root*, the name of the local machine must appear in the */.rhosts* file of the remote machine.



# Unix Backup Commands

- To perform a remote restore using the restore command, the operator can make use of the -f option again.
- f dump file: This option tells restore to use dump file as the file to restore from. If dump file is specified as a hyphen ( - ), restore reads from the standard input.
- NOTE: Because restore is also normally run by root, the name of the local machine must appear in the /.rhosts file of the remote machine.
- To copy a home partition to another machine with a remote restore, consider the following command, substituting the appropriate directory and device names.
- ```
# dump 0f - /dev/rdisk/c0t2d0s2 | (rsh machine ; cd /home;restore xf -)
```



# Unix Backup Commands

- **tar Command**
- **What happens if the operator does not want to dump and restore complete file systems?**
  - For example, what if a user simply wants to make a tape of the data associated with one project?
  - Most UNIX derivatives provide a standardized utility called *tar*. The *tar* command creates tape archives and provides the ability to add and extract files from these archives.

```
% tar c -C /home/project include -C /data/project
```

```
% tar cvf /dev/rmt/0
```

- Create a tar archive on device /dev/rmt/0

```
% tar tvf /dev/rmt/0
```

- List the contents of the tar archive on /dev/rmt/0

```
% tar xvf /dev/rmt/0
```

- Extract the contents of the tar archive on /dev/rmt/0



# Unix Backup Commands

- Another command that may be used to back up and restore files is *cpio*.
- The *cpio* command copies file archives “in and out.”
- The syntax for the *cpio* command follows.

`cpio -key [options] filename`

- The key option to *cpio* determines the actions to be performed. The following flags are mutually exclusive.



# Unix Backup Commands

- `% ls | cpio -oc >> ../newfile`
  - NOTE: The `find`, `echo`, and `cat` commands can also be used as substitutes for the `ls` command to produce a list of files to be included in `../newfile`.
- To extract the files from `../newfile`, issue the following command.
- `% cat newfile | cpio -icd`
- `% find . -depth -print | cpio -pdlmv newdir`
  - NOTE: When using `cpio` in conjunction with `find`, use the `L` option with `cpio`, and the `-follow` option with `find`.



# Unix Backup Commands

- Yet another command available to use as a backup-and-restore utility is the *dd* utility.
- The *dd* command copies the input file to output, applying any desired conversions in the process.
- When complete, *dd* reports the number of whole and partial input and output blocks.

`dd [ option = value ] ....`





# Unix Backup Commands

- Do not use `dd` to copy files between file systems with different block sizes.
- Using a blocked device to copy a file will result in extra nulls being added to the file, in order to pad the final block to the block boundary.
- When `dd` reads from a pipe, using the `ibs = X` and `obs = Y` operands, the output will always be blocked in chunks of size `Y`.
- When `bs = Z` is used, the output block size will be whatever could be read from the pipe at the time.



# Unix Backup Commands

- if = filename: Use filename as the input file. stdin is the default value.
- of = filename: Use filename as the output file. stdout is the default value.
- ibs = n: Use n as the input block size. 512 is the default value.
- obs = n: Use n as the output block size. 512 is the default value.
- bs = n: Use n as the input and output block size This supersedes the ibn and obn arguments. If no conversion is specified, preserve input block size.
- files = n: Copy and concatenate n input files before terminating.
- skip = n: Skip n input blocks before performing copy.
- isseek = n: Seek n blocks from the beginning of the input file before copying.
- oseek = n: Seek n blocks from the beginning of the output file before copying.
- count = n: Copy n input blocks.
- swab: Swap every pair of bytes.
- sync: Pad every input block to ibs.



# Unix Backup Commands

```
# dd if=/dev/rdisk/c1t3d0s2 of=/dev/rmt/2 bs=20b
```

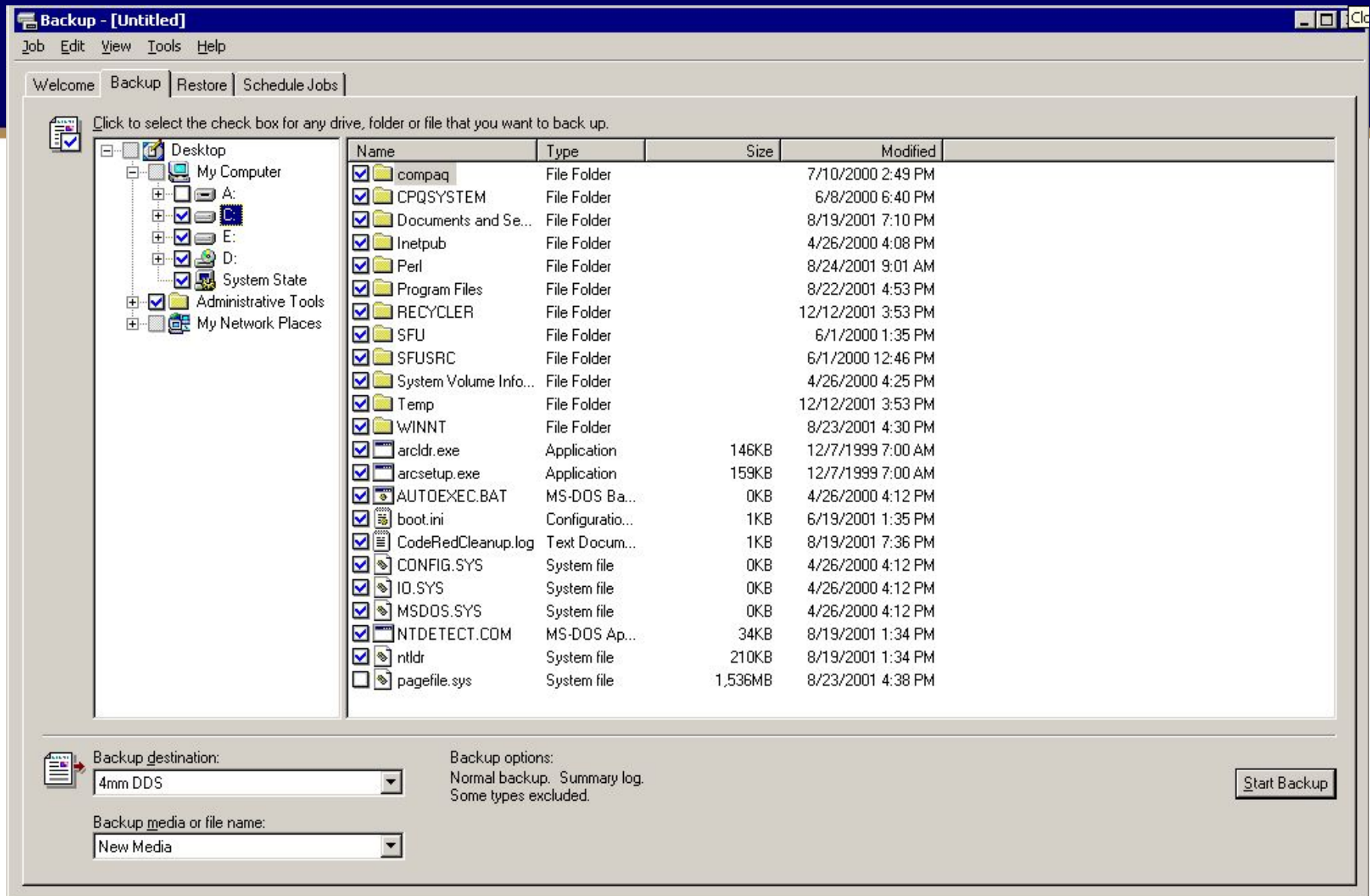
```
# dd if=/dev/rmt/2 of=/dev/rdisk/c2t1d0s2 bs=20b
```



# Windows Backup Commands

- Like its UNIX cousins, Windows provides a utility to perform file system backups.
- The Windows backup-and-restore utility (backup.exe) provides for backups, and restores.
  - In backup mode, the operator is given the ability to select which disk and/or files to back up, whether the local registry should be dumped, whether the backup should use compression, and what users may read the backup tape(s).
  - Otherwise, the “menu” of backup options available to the operator is pretty limited.

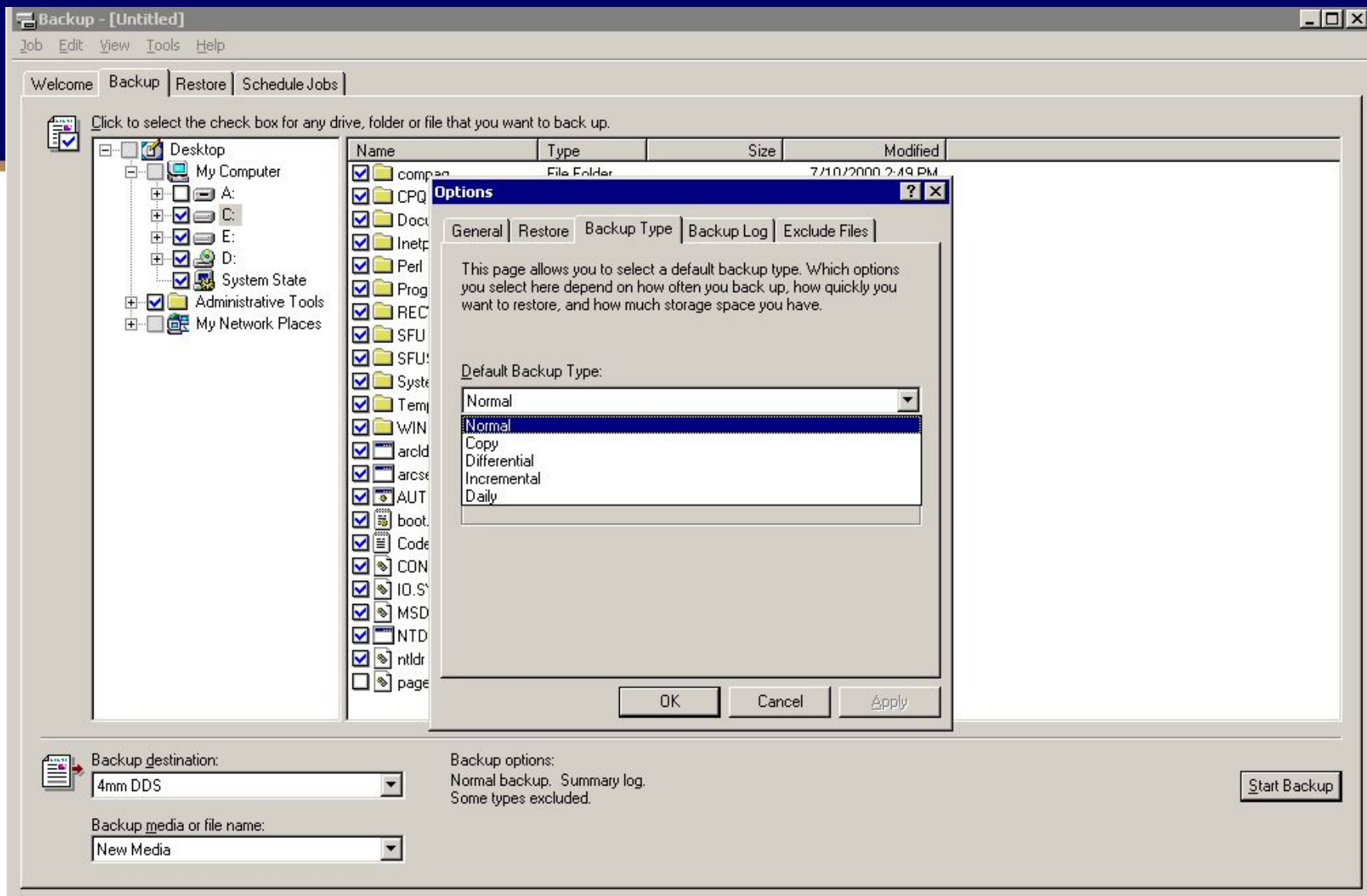




# Windows Backup Commands

- The Options menu under the Tools menu allows the operator to determine the type of backup to be performed, whether data should be verified after the backup is performed, the amount of detail supplied in the log files, whether new media should be used, backup scheduling, and other configuration parameters for the backup utility.



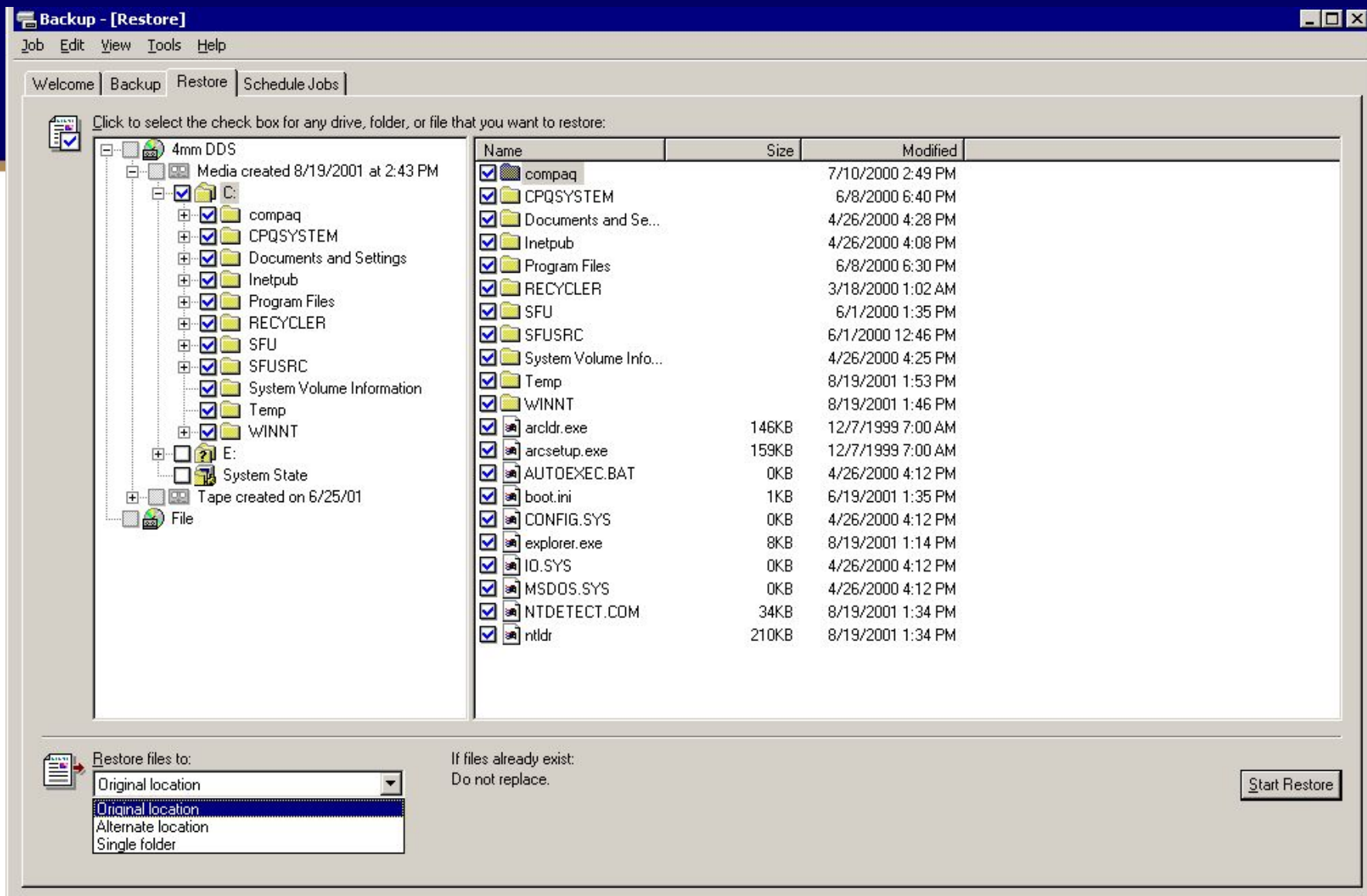


# Windows Backup Commands

- The Windows backup utility also provides the interface to the *restore* function.
  - In restore mode, the utility displays a catalog of the tape, allowing the user to select which files/directories need to be restored.
  - The operator is also given the choice of where the file is to be restored.
- Although the GUI may simplify the setup of Windows backups, it also limits the choices available to the operator. Because this utility is based on the backup utility offered in the consumer versions of Windows, the sysadmin at a commercial site may decide that the Windows backup utility is not the first choice for backup software at the site. Many third-party backup utilities are available for Windows systems, including Amanda, Legato Networker, and the Veritas backup suite.







# Dealing with Specific Backup Issues

- Certain aspects of successful backup and restore strategies require special attention.
  - For instance, how could the operator restore the root file system if the root disk had crashed and there was no way to boot the system?
  - Many administrators are also concerned with how to automate backups to minimize time investment while ensuring successful backups.
  - Next, what happens if a backup requires 2 Gb of backup media, but the backup device can write only 1 Gb to the media?



# Dealing with Specific Backup Issues

- Restoring the Root File System
  - One of the most difficult problems faced when using *restore* is restoring the root file system.
    - If the root file system is missing, it is not possible to boot the damaged system, and there would not be a file system tree to restore to.
    - One way to accomplish a root file system reload is by booting the system to the single-user state from the CD-ROM distribution media.
    - Another way to reload the root file system would be to boot the system to the single-user state as a client of another system on the network.
    - Another method of restoring the root file system is to remove the disk from the system, and attach it to a working system.



# Dealing with Specific Backup Issues

- Multi-volume Dumps

- Two of the backup commands mentioned in this chapter also allow for multi-volume backups.
  - The **dump** command and the **cpio** command allow a backup to be stored over multiple media sets.
  - The other commands (**tar** and **dd**) will allow the user to split the backups onto several sets of media, but these commands require that the user perform much of the work manually.
  - The **cpio** command watches for the “end of medium” event. When **cpio** detects this event, it stops and prints the following message on the terminal screen.

If you want to go on, type device/file name when ready.

- To continue the dump, the operator must replace the medium and type the character-special device name (e.g., */dev/rdiskette*) and press Enter. At this point, the operator may choose to have *cpio* continue the backup to another device by typing the device’s name at the prompt.



# Dealing with Specific Backup Issues

- Multi-volume Dumps
  - Using *dump* as the backup command is somewhat simpler.
    - Like *cpio*, *dump* will detect the end-of-medium event and stop operation.
    - The *dump* command will then wait for the operator to change the medium before it continues.
    - Unlike *cpio*, *dump* does not need the name of the backup device to continue.
    - The operator simply needs to confirm that the medium has been changed and that everything is ready for *dump* to continue operation.



# Automated Backups

- Cheap and Dirty
- `#!/bin/sh`
- `# dump /, /usr, /var, /home, /opt, /orasoftware`
- `#`
- `mt -f /dev/nst0 rewind`
- `dump 0bdfu 50 54000 /dev/nst0 /dev/ida/c0d1p2`
- `dump 0bdfu 50 54000 /dev/nst0 /dev/ida/c0d2p1`
- `dump 0bdfu 50 54000 /dev/nst0 /dev/ida/c0d3p1`
- `dump 0bdfu 50 54000 /dev/nst0 /dev/ida/c0d4p1`
- `dump 0bdfu 50 54000 /dev/nst0 /dev/ida/c0d5p1`
- `dump 0bdfu 50 54000 /dev/nst0 /dev/ida/c0d1p3`
- `mt -f /dev/nst0 rewind`
- `mt -f /dev/nst0 offline`
- `# sh -x /dodump`



# Cross Platform Backups

- Several utilities are available for cross-platform backups:
  - Veritas
  - Legato Networker
  - RaxCo
  - Amanda – free – <http://www.amanda.org/>



# Summary

- This chapter explored the commands that can be used to make backup copies of system data, why it is important to make such backup copies of data, and selected methods of avoiding data loss due to natural or other disasters.
- The authors hope that readers never have to use any of these backup copies to restore the operation of their systems, but such restorations are inevitable.
- Good backups require a lot of time and attention, but having a reliable copy of data is much more acceptable than the time and expense of rebuilding a system without such backup copies.





- Unix And Linux System Administration Handbook, 5th Edition By Evi Nemeth, Garth Snyder, Trent R. Hein, Ben Whaley, Dan Mackin. Released September 2017. Publisher(s): Addison-Wesley Professional. ISBN: 9780134278308
- Practice Of System And Network Administration, The: Devops And Other Best Practices For Enterprise IT, Volume 1, By Thomas A. Limoncelli, Strata R. Chalup, Christina J. Hogan. Released November 2016. Publisher(s): Addison-Wesley Professional. ISBN: 9780133415087
- Essential System Administration, Third Edition by Æleen Frisch, Published by O'Reilly Media, Inc. (2008) ISBN: 0-596-00343-9

