BackUP

COLLABORATORS

	TITLE : BackUP		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		November 24, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

BackUP

1.1 BackUP.guide

```
*****                 **     **   ** *****
**  **                **     **   ** **   **
**  **  ****    ****  **  ** **   ** **   **
*****      **  **  ** ** **  **   ** *****
**  **  *****  **      ****   **   ** **
**  ** **  **  **  ** ** **  **   ** **
*****   *** **  ****  **  **  ****  **
```

1.2 legal stuff

```
-----------------------------------------------------------------------------
****************************** Legal Stuff *********** September 16, 1993 *
-----------------------------------------------------------------------------
```

BackUP is a shareware, freely distributable hard drive backup program for the Amiga under Workbench 2.x. If you like BackUP and regularly use it, I would appreciate being sent a $20 contribution to the following address:

Felix R. Jeske
3746 North Oleander Avenue
Chicago, IL 60634-3210
USA

Usenet: fjeske@amiganet.chi.il.us

Contributors will receive the latest version of BackUP (I am already adding a few other goodies) plus a few other programs I’ve written but not published.

Suggestions, comments and criticisms (ouch) are also welcome at the above address or on Usenet. I am quite proud of BackUP and gladly support registered users. If any problems are encountered, PLEASE report them! The fastest way to get a problem report to me is by leaving e-mail at the above Usenet address. I have sent out disks at my own expense to users who report problems to correct them.

DISCLAIMER

FELIX R. JESKE MAKES NO WARRANTIES EITHER EXPRESSED OR IMPLIED, WITH RESPECT TO THIS SOFTWARE, ITS QUALITY, PERFORMANCE OR FITNESS FOR ANY PARTICULAR PURPOSE. THIS SOFTWARE IS PROVIDED "AS IS." THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOFTWARE IS WITH THE USER. IN NO EVENT WILL FELIX R. JESKE BE LIABLE FOR DIRECT, INDIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT IN THE SOFTWARE.

```
----------------------------------------------------------------------------
****************************************************************************
----------------------------------------------------------------------------
```

1.3 credits

```
----------------------------------------------------------------------------
********************************* Credits **********************************
----------------------------------------------------------------------------
```

Many thanks go to the developers of ARP who have created an extremely useful collection of routines in one library and have made it easy to program with as well.

Thanks go to Holger P. Krekel and Olaf Barthel for use of their lh.library. This is an excellent version of the LZH compression algorithm that, again, is very easy to program.

Thanks to Ludwig Kamphenkel for his translation of the text in BackUP into the German.

Finally, thanks to Urban D. Müller, et al. for XPK. XPK is a great collection of packers and encrypters.

```
----------------------------------------------------------------------------
****************************************************************************
----------------------------------------------------------------------------
```

1.4 system requirements

```
----------------------------------------------------------------------------
*************************** System Requirements ****************************
----------------------------------------------------------------------------
```

BackUP requires Workbench 2.x, 1MB of RAM, a hard drive (obviously) and as many floppy drives as you can afford. BackUP requires lh.library V1 (available on FF436, distributed with BackUP).

```
----------------------------------------------------------------------------
****************************************************************************
----------------------------------------------------------------------------
```

1.5 historical info

```
----------------------------------------------------------------------------
***************************** Historical Info ******************************
----------------------------------------------------------------------------
```

BackUP was initially developed on an A500 under Aztec C 3.6a. Workbench 2.x additions were made on an A3000 under Aztec C 5.2a. Development has since shifted to an A4000/040 under SAS/C 6.3.

```
  V3.91  -  XPK support added, fixed sensing of new (3.0) disk formats

  V3.90  -  Add pseudo-localization in the form of a separate German
            version of program.  Lots of other little fixes, tweaks and
            twiddles.

  V3.89  -  Added alphabetization option to file requestor, worked around a
            SAS/C 6.2 bug regarding inlined memcmp() and started internal
            restucturing for a localized version

  V3.88  -  Added safe backups and alternate restore destination, fixed
            recognition of RAD devices as backup partitions and modified the
            backup log name

  V3.87  -  Fixed high-density floppy support (now that I got an A4000)

  V3.86ß -  Added ability to backup to another hard drive partition
            (not fully supported yet so current off)

  V3.85  -  Added don't compress pattern and changed highlighting in file
```

```
          requestor

 V3.84  -  Added recursive selection, fixed directory state bug and
           removed 6.5 MB file limit

 V3.83  -  Fixed file scan array resizing bug

 V3.82  -  More optimizations and changed file requestor behavior

 V3.81  -  Removed internal sound generator for bell, used 2.1 DisplayBeep()

 V3.80  -  Discontinued use of arp.library

 V3.79  -  Changed manner in which disk drives are scanned for

 V3.78  -  Fixed color palette bug introduced during optimization frenzy

 V3.77  -  Converted array indexing to pointers to save code

 V3.76  -  Fixed maximum number of files (2000) on restore problem

 V3.75  -  Fixed internal tracking of bad tracks

 V3.74  -  Added support for hard file links

 V3.73  -  Added optional bell on message and disk change requestors

 V3.72  -  Fixed disk labeling bug

 V3.71  -  Fixed directory selection bug and improved selection overall

 V3.70  -  Improved input-blocking code for multiple requests

 V3.69  -  Added status (Reading..., Writing..., etc.) display

 V3.68  -  Rearranged menu system

 V3.67  -  Added two different types of backup logs

 V3.64  -  Squashed file protection bug

 V3.63  -  Squashed empty file bug

 V3.62  -  Squashed CrossDOS incompatibility

 V3.61  -  Squashed drive motor bug

 V3.6   -  Added color requestor and support for high density drives

 V3.5   -  First public release

 V3.4   -  Added compression using lh.library

 V3.0   -  Discontinued use of req.library and added custom gadgetry

 V2.0   -  Added gadgets using req.library
```

```
  V1.0   -  Added primitive interface using autorequestors and the console

  V0.9   -  Initial development of the backup and restore engines
```

I started developing BackUP after my purchase of a $700 (groan) 30MB Supra hard drive. Not wanting to spend any more money than I had to, I started on BackUP after reading about programming the trackdisk.device in an issue of Transactor for the Amiga by Bob Rakosky (August '89: Vol. 2, Issue 5). After figuring out how to do raw reads and writes to the floppy drive, I learned about parsing a partitions directory structure. I heard about ARP in another issue of Transactor and finally got my hands on it. I incorporated req.library after buying CygnusEd which uses it quite heavily. The current version of the code is almost identical to that under req.library except for cosmetics. I borrowed the current gadgetry style from AVS (Application Visualization System), a scientific visualization package I program and use at work on UNIX workstations. Finally, I found lh.library on Fred Fish Disk #436 and included compression as an option.

```
----------------------------------------------------------------------------
****************************************************************************
----------------------------------------------------------------------------
```

1.6 benchmarks

```
----------------------------------------------------------------------------
******************************** Benchmarks ********************************
----------------------------------------------------------------------------
```

The target machine for BackUP is any Amiga with at least two (2) floppy drives. Being restricted to one floppy drive eliminates the continuous write feature. BackUP does not support tape drives since I don't have one (yet!). Compression is only recommended for fast machines or if you really want to save disks (190% compression is typical) as the on-the-fly compression does slow down the backup substantially. Compression actually speeds up the restore since the speed bottleneck is the floppy read/write time. The decompression process is so fast that reading in less data, decompressing it in memory and writing it out to the hard drive is faster than reading and writing out the original uncompressed data which took longer to load from the floppy.

My only real comparison of the capabilities of BackUP to other hard drive backup programs is my experience with HDBackup (shipped with my A3000) and other PD backup programs I've uploaded. Since most of the PD backup programs I've obtained are CLI based, I only compare BackUP and HDBackup since the CLI is not well suited to perform such an operation. Therefore, the following is a comparison of BackUP and HDBackup working on 294 files comprising about 1.1MB of data with compression and verify on. It shows the following:

	BackUP	HDBackup
Executable Size	40K	81K
Memory Usage	430K	475K

While Backing		
Time to Backup	2:56	4:03
Number of Disks	1	3

I could not figure out why HDBackup needed three (3) disks to backup 1.1MB of data, especially when full compression was enabled. In uncompressed form, it should have only occupied two disks.

****

1.7 user documentation

*************************** User Documentation *****************************

Backing Up

Restoring Backup

Recovering Backup

Compression

****

1.8 backing up

******************************* Backing Up *********************************

The backup procedure consists of selecting the partition to backup, the specific directories and files to backup within the partition, selecting which floppy drives to use during the backup and finally the swapping of disks in and out of the floppy drives.

When BackUP boots up, it polls the machine for all hard drive partitions and all floppy drives. A gadget for each is created on the main screen. The user need only press on one of the labeled partition buttons to start reading the complete directory contents into memory. When done, a standard file requestor displays the entire directory structure.

A few methods are available to the user as to how files are selected to be included in the backup. The order of the methods chosen is important. For example, selecting a particular file in a directory and then deselecting the entire directory will deselect that file. The following lists the order in which the selection process should proceed.

1) The Incremental/Full button changes whether files with their archived bits set are included in the selected file list.

2) The Include and Exclude wildcard patterns are convenient ways to include/exclude large groups of files. A file is selected for backup if it matches the Include pattern and does NOT match the Exclude pattern. An ARP compatible pattern matching system is used and therefore asterisks (*) can be used in place of Commodore’s global wildcard (#?). Also patterns can be ORed together via the pipe (|) operator to form more complex pattern such as:

   ```
   *.(c|h)|*.doc
   ```

   which would match all files ending with either a .c, .h or .doc extension.

3) Whole directories trees can be manually included or excluded by single clicking on them in the file requestor with the recursive select option on. The immediate contents of a directory may be selected if this option is off. Directory names with a highlighted background indicate that all the files in that directory are selected. Double clicking on a directory name changes to that directory. All non-empty directories have a much-greater-than sign (») appended to their name denoting that they may be entered.

4) Finally, individual files can be selected (unselected) by single clicking on them in the file requestor.

After choosing the files to be backed up, the user can press the buttons for each of the floppy drives BackUP will use during the procedure. It is recommended that as many drives be used as available. BackUP automatically formats disks as it writes and also switches between drives without any user intervention (continuous write). This speeds the process up by constantly writing to one drive while the user is changing the disk in another.

Finally, a number options may be set in the menu bar: compression, verify, logs, beeping and the palette. BackUP performs on-the-fly compression by reading in small (11K) blocks of files, compressing them and asynchronously writing them to disk. The asynchronous part allows BackUP to read from the hard drive and write to the floppy drive simultaneously. Compression can reduce the number of disks used by a factor of two (I have seen 2MB written to 1 disk). Compression does, however, slow down the backup since the compression process takes time and degenerates the backup to a synchronous process (read, compress, write).

The compression option has been extended in BackUP to include XPK support. XPK is an external set of libraries that perform compression and data encryption. Now, in addition to the LhLib compression scheme that was implemented in V3.4, the user has the choice using one of the XPK compressors instead. This feature has been implemented in such a way that BackUP automatically detects the type of compression on restore and calls either LhLib or XPK appropiately. Note that previous versions of BackUP cannot uncompress the XPK compressed backups. See the "Compression" section below regarding benchmarks for the various compressors.

The verify option specifies whether a read pass of the tracks written to the floppy is made after the format/write pass. The backup process is sped up considerably with verify off, however, it is not recommended since the integrity of the data is unknown.

Two types of backup logs are available: sortable and printable. The sortable type simply lists the file, its path, creation date, size and backup status in tab separated columns that can be passed onto other codes to be sorted at will. The second backup log type lists by directory the same information. As the name suggests, it is more appropriate to be printed for future reference. The name of the backup log file can be chosen when either type log file is selected. The current backup partition name can be put into the log name by placing a "%n" somewhere in the name. The current date may be automatically put into the name (in DD-MM-YY format) by placing a "%d" somewhere in the name. Note also that this name is only a root name and that either ".inc" or ".ful" is appended to the log name depending on the backup type.

The beep option allows the user to choose whether or not an audible beep is made when a pop up message and/or a disk request is made. This is useful if the user walks away from the machine during the backup or restore process.

Finally, the color palette may be changed to suit the user’s preferences. As mentioned above, the gadgetry style was modeled off of AVS. The default color map was chosen to enhance the 3D feel of the interface.

All of the options (include/exclude wildcards, compression state, verify state, backup type, etc.) can be saved as the defaults by choosing the "Save Configuration" option in the Options menu. Upon subsequent boot ups, BackUP will load the configuration file (S:BackUPDefaults) and automatically set the previously saved defaults. The configuration file should be compatible between BackUP versions as it is always appended to. If, however, BackUP boots up with "funky" colors (as sign of configuration file incompatibility), just delete the file, reset your options and resave the configuration.

After all this is completed, the user need only press the "Start Backup" button and begin swapping disks. During the backup, a fuel gauge bar fills representing the completion of the backup. Also, the current file being backed-up as well as the current disk being written to and the total number of files and bytes backed-up are constantly updated. The disks should be labeled by date and disk number since BackUP will request numbered disks during the restore procedure. The user may halt the process at any time by pressing the "STOP" button. Note that the process will not actually stop until the current file is completely written.

After the files are copied to floppy, BackUP writes out the directory structure. Only the directories that contain backed-up files are written in order to save space. This means that, on full backups, empty directories will not be saved, and, therefore, cannot be restored.

When backing-up, the "Safe Backup" feature can be enabled via the "Options" menu to protect the user from a corrupted or destroyed directory information disk. This feature writes this information along with the backup data allowing all the disks to be rescanned in case the directory information is otherwise lost. This feature has been implemented in a backward-compatible fashion and only costs 1-2% extra disk space.

```
----------------------------------------------------------------------------
****************************************************************************
----------------------------------------------------------------------------
```

1.9 restoring backup

```
----------------------------------------------------------------------------
**************************** Restoring Backup ******************************
----------------------------------------------------------------------------
```

The restore procedure is very similar to the backup procedure with the exception that the list of files is read from the floppy set instead of the hard drive.

When BackUP boots up, the user need only press the "Start Restore" button to have BackUP read a partition’s directory structure off floppy. BackUP will request that the last disk of a backup set be inserted into any available drive to start this process. BackUP may ask that the previous disk be inserted depending if the directory structure write overlapped multiple disks. When finished, the user can select and deselect files just as in the backup procedure with the exception that the Incremental/Full button is not available.

After choosing which files to restore the user should press the "Start Restore" button again. The same window with fuel gauge bar will appear and BackUP will request particular numbered disks to be inserted in the available drives. Files are restored to their previous location with date stamp, file note and protection bits restored as well. Directories will be made if necessary.

Files may be restored to an alternate destination if one is specified via the "Options" menu. This features strips off the leading volume name of the file’s path and replaces it with specified alternate destination which must contain a volume name and optionally a path (e.g., vol:path1/path2 is OK but path1/path2 is insufficient).

```
----------------------------------------------------------------------------
****************************************************************************
----------------------------------------------------------------------------
```

1.10 recovering backup

```
----------------------------------------------------------------------------
**************************** Recovering Backup *****************************
```

```
----------------------------------------------------------------------------
```

The "Safe Backup" feature writes extra directory information along with the data that is backed-up. This is in case the directory information written at the end of the backup set becomes corrupted which would previously render the entire backup set useless. In case this occurs, the "Recover Backup" feature is used to rescan all the disks in order to obtain this critical directory information.

The process begins by selecting "Recover Backup" and sequentially swapping all the backup disks in the floppy drives of the particular backup set. This builds an internal directory structure similar to the restore process above. The differences are limited to the fact that file comments and links are not kept track of and are therefore lost when recovering a backup. All other information is preserved: file name, path, size, date stamp and protection bits. Once all the disks have been read, the file requestor is updated and a normal restore process can continue.

Note that once everything is restored it is HIGHLY recommended to perform another backup since recovering a backup is a very inefficient way to perform a restore.

```
----------------------------------------------------------------------------
****************************************************************************
----------------------------------------------------------------------------
```

1.11 compression

```
----------------------------------------------------------------------------
****************************** Compression *********************************
----------------------------------------------------------------------------
```

With the addition of XPK support, the user now has quite a number of choices regarding the type of compression performed during the backup process. Six compressors are now distributed with BackUP. Other XPK compressors are available (e.g, RLEN), however, they are only demonstration compressors and are not useful in real applications. Each of the compressors have different characteristics in terms of speed and compression ratio (defined as the ratio of original data to compressed data).

The following list some benchmarks I have made on various data. The "Time" column lists the number of seconds to perform the backup on my A4000/040. The "Compression Ratio" column lists the compression ratio for the backup. The "Ratio" column lists the normalized (by None) ratio of "Compression Ratio" by "Time". This gives an indication of the tradeoff of time and compression ratio. In parentheses, next to each column, is the rank of the compressor in each of the three catagories with the sum being listed in the last column. All this is done because it is interesting to know how compressors compare in terms of speed, compression ratio, the tradeoff of these and finally, all of three considered as a whole. The point being the compressor with the lowest sum (highest ranks) is probably optimal.

```
        77 Files - 1483351 Bytes - Mixture of binaries, ascii, etc.
```

	Time [s]	Compression Ratio [%]	Ratio	Sum
None	130 (7)	100 (8)	1.00 (8)	23 (7)
LhLib	131 (8)	224 (2)	2.22 (5)	15 (5)
BLZW	77 (2)	178 (5)	3.01 (2)	9 (2)
HUFF	87 (3)	153 (6)	2.29 (4)	13 (4)
IMPL	119 (5)	202 (4)	2.21 (6)	15 (5)
NUKE	72 (1)	203 (3)	3.67 (1)	5 (1)
RLEN	99 (4)	130 (7)	1.71 (7)	18 (6)
SHRI	123 (6)	232 (1)	2.45 (3)	10 (3)

28 Files - 1686548 Bytes - DEM files from VistaPro

	Time [s]	Compression Ratio [%]	Ratio	Sum
None	145 (7)	100 (8)	1.00 (8)	23 (8)
LhLib	122 (4)	159 (2)	1.89 (3)	9 (3)
BLZW	102 (1)	145 (3)	2.06 (1)	5 (1)
HUFF	103 (2)	143 (4)	2.01 (2)	8 (2)
IMPL	133 (5)	138 (6)	1.50 (6)	17 (6)
NUKE	111 (3)	141 (5)	1.84 (4)	12 (4)
RLEN	143 (6)	102 (7)	1.03 (7)	20 (7)
SHRI	152 (8)	162 (1)	1.55 (5)	14 (5)

492 Files - 1491036 Bytes - Ascii texts files (#includes from SAS/C)

	Time [s]	Compression Ratio [%]	Ratio	Sum
None	136 (6)	100 (8)	1.00 (8)	22 (6)
LhLib	138 (7)	252 (2)	2.48 (2)	11 (2)
BLZW	116 (2)	179 (5)	2.10 (5)	12 (3)
HUFF	118 (3)	130 (6)	1.50 (6)	15 (4)
IMPL	128 (4)	210 (4)	2.23 (4)	12 (3)
NUKE	111 (1)	225 (3)	2.76 (1)	5 (1)
RLEN	133 (5)	104 (7)	1.06 (7)	19 (5)
SHRI	149 (8)	257 (1)	2.35 (3)	12 (3)

From the above, it can be gleaned that SHRI is the best compressor (i.e., highest mean compression ratio), NUKE and BLZW are the fastest compressors, NUKE has the best tradeoff in terms of compression ratio to speed and NUKE and BLZW are the best overall. A future release of BackUP will offer two more compression options ("Minimum Disks" and "Minimum Time") that use statistical data such as above in order to create a backup using the minimum number of disks (without regard to time) or in the minimum amount of time (without regard to disks used) by automatically sensing the file type (ascii, executable, data, etc.) and choosing an appropriate compressor.

```
----------------------------------------------------------------------------
****************************************************************************
----------------------------------------------------------------------------
```

1.12 known problems

```
----------------------------------------------------------------------------
**************************** Known Problems ******************************
----------------------------------------------------------------------------
```

There are few known problems with the program. The first is that although a "Cancel" button is included on a number of popup requestors, it is not always implemented and therefore pressing it just brings up the same requestor. This is true only for requestors with red text (as opposed to white) that appear in the middle of screen that usually request that a disk be (re)placed in a drive (usually, truly canceling the request will destroy the backup or restore process).

This is not really a problem but a known bad interaction. If BackUP is run with BlitzDisk (V2.00 © 1989 Microsmiths) caching a hard drive partition, the file scan of that partition is slowed down substantially (the other portions of the code don't seem to be affected). I can only figure that both processes are contending for the CPU and therefore BackUP visibly slows down. If anyone has any ideas as to a fix for this, I'd be happy to hear it.

A problem has cropped up between versions prior to 3.73 and those after. In order to support hard file links, the format of the backup on floppy changed. This, of course, makes versions 3.74 and on completely incompatible with versions earlier than this. Fortunately, not too many people have these versions.

Due to the increasing number of hard drive cache programs, a warning must be added about the use of them during a backup. It is STRONGLY recommended that such programs be turned OFF prior to backing-up a partition. These programs usually operate at the driver level (well below BackUP) which prevents any detection of their unseen operation. BackUP expects tracks to have been written to or read from disk, however, with a cache, they are written to or read from cache buffers. This can confuse BackUP and prevent its normal operation (as evident from "ERROR: Bad Disk! ..." requestors that pop up randomly on newly inserted disks). BackUP rigorously invalidates buffers prior to all reads and flushs all buffers after a write, however, this may not be enough for some cache programs. Hypercache Pro (V1.01B © Dave Plummer), in particular, is known to be fully compatible with BackUP.

```
----------------------------------------------------------------------------
****************************************************************************
----------------------------------------------------------------------------
```

1.13 release notes

```
----------------------------------------------------------------------------
**************************** Release Notes *******************************
----------------------------------------------------------------------------

Release                                 Comments

 V3.91      This version adds support for XPK packers and encrypters.  This
```

version also fixes the problem with sensing the new (3.0) disk formats (DOS2 - DOS5). These include the new "International" and "Directory Caching" formats.

V3.90 This version is now available in German. The internal changes required for this include converting to the GadTools system of menus as well as removing hard-coded text positions. In addition, quite a number of other little fixes, tweaks and twiddles have been made. For example, more strict buffer invalidation and disk updating (to thrwart cache programs), slight position changes to the main-screen buttons and other small, internal changes.

V3.89 This version includes a workaround for a SAS/C 6.2 bug in the inline memcmp() routine. The code compared one byte beyond the length of the specified buffer and randomly caused "BAD DISK!" requestors to pop up. This version has a hand-coded memcmp() routine optimized for the particular comparison made. Also included in this release is some internal restructuring made necessary for localization of the menu and requestor text. Version 3.90 should be available in German as well as English.

V3.88 This version adds "safe backups" allowing disks to be rescanned via the "Recover Backup" item in case the directory information written at the end of the backup set becomes corrupted. Also, an "Alternate Destination" item has been added in the "Options" menu allowing files to be restored to an alternate destination instead of their original path. In addition, RAD type devices are no longer recognized a valid backup partitions. Finally, the backup log name accepts %d and %n specifications for date and partition name, respectively. Previously, just a %s was used for the date and, if two or more partitions were backed-up on the same day, the log files would get clobbered (oops!).

V3.87 This version fixes the high-density floppy drive support in the program. As I figured, this feature was broken previously but I could not fix it since I did not own a HD drive. Now that I am developing (and backing up) on an A4000, I fixed this. The fix was not simple because queries needed to be placed at numerous points in the code to determine what type of disk was in the drive at the time of the query.

V3.86ß This version adds the capability to backup to another harddrive partition. However, this capability has been hacked in so far and therefore is not a fully supported feature.

V3.85 This version adds a pattern against which files are compared to see if they should be compressed. This speeds up the backup process since files that have been already compressed (with *.lha, *.lzh, *.zoo, *.arc and *.dms extensions, for example) usually do not compress any further and therefore only slow down the backup process. Also, files are now highlighted by changing their background which is consistent with how directories are highlighted. File names are always blue (in the default color scheme, anyway) and directories are always red.

V3.84 This version adds optional recursive selection in the file requestor. This means that clicking on directories (de)selects everything immediately in them and in any subdirectories and so on. Also, a bug with the proper highlighting of directories was fixed. Finally, an internal limit of 6.5 MB for a single file that could be backed-up was removed.

V3.83 This version fixes an "off by 1" bug that limited the number of files that could be scanned in off a hard drive. Similar to the bug fix for V3.76, this time, however, the problem occured just after the files were read in off the hard drive and initially displayed in the requestor. The "off by 1" comes in because BackUP resized an array one element after it exceeded the size of the array (not one element before) and this trashed some random memory.

V3.82 This version includes more optimizations that cut code size and includes a requested change to the behavior of the file requestor. It now keeps track of where (i.e., position in the file list) a user was when he enters a new directory and returns him to the same place. This eliminates the hassle of scrolling back to where you were after pressing the "Parent" button.

V3.81 This version removes internal sound generation code in favor of DisplayBeep(). Prior to 2.1, DisplayBeep() only flashed the screen. Now, however, a user specified sound and/or a screen flash can be generated at the user’s preference.

V3.80 This version discontinues the use of ARP completely. All the functions BackUP used from ARP were in AmigaDOS 2.x so they were simply converted. Also, ARP is stagnant to my knowledge so the functions in AmigaDOS should be more up to date.

V3.79 This version replaces the ARP calls that scan for disk drives (hard and floppy) with AmigaDOS versions of the same. The routine should now be fool-proof (yeah right) against CrossDOS devices calling up requestors. Also the hard drive volume names are used instead of their device equivalents which is more consistent with other parts of the code.

V3.78 This version fixes a small bug in the color palette requestor. BackUP updated the numbers above the slider gadgets with the values of color 0 instead of the appropriate color number.

V3.77 This version includes some optimizations of array addressing that were converted to pointers. For example, tracks[numtracks++] was converted to *tracks++. This and other small tweeks led to 1100 bytes of code being saved.

V3.76 This version fixes a bug concerning the maximum number of files that can be restored. In order to save memory, BackUP dynamically allocates buffers to some initial size, then, whenever they are about to be exceeded, they are resized. BackUP forgot to resize the file buffer (currently set to 1000 files). It was resized correctly during the backup procedure so an unlimited number of files can be backed up.

V3.75 This version fixes a long-standing bug that screwed up internal tracking of bad tracks on backup floppies. If a track refuses to verify, internal updates are necessary to reflect the new track numbers to which file data has been moved to. Due to the asynchronous operation, this becomes complicated since data is buffered and verification of data lags behind writing of data. Put simply, BackUP would determine track n-1 is bad when it is ready to write out track n so it would have to move track n-1 data to wherever and track n data to wherever+1.

V3.74 This version adds support for hard file links. The link is examined to determine where it is pointing to and this information is save. On restore, the link is restored via MakeLink(). This, however, requires that the destination file be present to be successful. BackUP, therefore, restores all real files first and then restores file links. If the user did not restore the destination file or else intentionally deleted it, the MakeLink() will fail and the user will be informed. Soft links are not yet supported. Directory links are ignored.

V3.73 This versions adds an optional audible bell cue when either messages pop up or a disk change is requested.

V3.72 This version fixed a bug in the labeling of disks. Proper error checking was not performed to ensure that the disk label was actually written. If a user popped out a disk during the labeling process, the label would not be written and this would corrupt the backup set.

V3.71 This version fixed the directory selection bug in which BackUP lagged in noting when a directory was selected (i.e., all the files in the directory were selected). In addition, the background color of a selected directory is changed to note its state.

V3.70 This version improves the manner in which input is blocked to windows when they are superseded by another request. Earlier versions used a cumbersome system that failed to work on multiple, overlapping user requests.

V3.69 This version adds a flashing status display informing the user of the exact operation being currently performed (Reading..., Writing..., Compressing..., Decompressing... and Verifying...). Note that some of the messages flicker so fast that they sometimes cannot even be seen.

V3.68 This version reorganizes the menu system by adding a second menu that is just for the growing list of backup options. The first menu is a standard Project menu.

V3.67 This version adds two types of backup log files: sortable and printable.

V3.64 This version removes a bug with read protection on a file. A proper check was not made that an error did not occur during the read of a file (as opposed to opening it which was checked). This failure to check the read status also caused a problem if a file was removed during the backup procedure.

V3.63 This version removes a serious bug with empty files. Backups made prior to this release cannot restore empty files as they crash the machine (oops!). As a work around, users should remove empty files from the restore list prior to restoring. I found this one restoring my 50MB Quantum onto my new 236MB ST3283N. I also removed a few potential divide-by-zeros if the user backs up zero bytes of files.

V3.62 This version removes an incompatibility with CrossDOS. The DIx: and PCx: devices where found by BackUP which called up a system requestor.

V3.61 This version actually squashes two bugs 1) the drive motor was left running when the user was informed that the current disk was write protected and 2) a tab creeped into one of the messages which printed as a square block in BackUP.font.

V3.6 This version contains two additional features as per Olaf Barthel's request: a color requestor and support for high-density drives. I am uncertain of the latter since I do not own such a drive and cannot, therefore, verify my implementation. I would appreciate user feedback on this. Questions I have concern what happens when a user puts a low-density disk into a high density drive? I am not certain I handle this case correctly since the code I inserted queries drive geometry not disk geometry and I don't think a DD disk can be formatted as HD.

V3.5 Original public release.

****
