00611d88-0

Timo Kaikumaa

COLLABORATORS

	TITLE : 00611d88-0		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Timo Kaikumaa	February 12, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

00611d88-0

1.1 UnixDirs3 - The Front Page

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

                        UnixDirs3 V1.5  (21.2.1997)

           Patches DOS to use Unix style paths  (OS3+ required)

                           Freeware, finally!!!

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

Have you ever used Unix systems (or MS-DOS, although no ordinary people can stand it for long) you may have noticed that the path specifications are not the same as they are in the Amiga. Not saying they should be, but it’s rather annoying to ask your Amiga to "CD .." in an out of habit fashion and see only an error message appearing. And of course, sometimes pure Amiga paths like "///foobar" may look like a bit confusing.

Fortunately, these were problems for me and so I wrote UnixDirs3. This tiny program allows you to use both Amiga and Unix style paths simultaneously. Other than that, there is a feature called multiparent. Now commands like "CD ..." and "CD ...." will change current directory level upwards more than one step. (Free hint: just using "..." and "...." seem also work thanks to console device.)

```
UnixDirs3 was developed on an ECS500 workstation, boosted by SupraTurbo28,
running OS3.1, equiped with 2+8 megabytes of memory and 1.5 gigs of hard disk
storage plus 4$\times$CDROM drive.
```

1.2 UnixDirs3 - Introduction

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

                              Introduction
---------------------------------------------------------------------------


Since this guide file is fairly large, I’ve highlighted the main areas in case
you have absolute no idea what the program does.

UnixDirs3 is similar program than its predecessors UnixDirs (by Murray Bennett
and Mark Cyster) and UnixDirsII (by Martin Scott). Other than providing the
same function, this version has nothing to do with them.

UnixDirs3 was written because the original UnixDirs wouldn’t even work with
OS2+ and the second attempt (UnixDirsII) has some nasty bugs in it. Bugs no
longer exist (at least the same bugs) and the routines now take care of all
OS3 path handling functions including file notifications. There is also this
new feature called multiparent...

So, if you still wonder what will you gain using UnixDirs3, here is a brief
summary (surely it has something to do with DOS paths). This is how the path
handling routines get modified:


1) A leading "." (period) means the calling task’s current directory. This
   works both alone (just the period) and at the beginning of the path (like
   "./subdir". So if you would like to copy a program called "C:List" to your
```

current directory you could write "Copy C:List ." or "Copy C:List ./".

2) Two periods (".." or "../") mean backing up one level in directory hierarchy. So, when referring to parent directory of "S:" (usually it's the boot directory or "SYS:") you could write "S:.." instead of S:/". More complex example of this would be to use "./../../filename" or simply "../../filename" instead of "//filename". As you can see, the standard Amiga paths are more compact, but perhaps not so readable.

3) With multiparent option activated it's now possible to back up more than one level (actually any number of levels) by using more than one subsequent periods. So the strange(?) path specification "///foobar" found in the
main~section
and its hopefully more readable counterpart
"../../../foobar" can be replaced with "..../foobar" meaning a file or directory called "foobar" located three levels upwards from your current directory position. Note how slashes in this new styled path are only separating different path components. In standard Amiga syntax they could also mean parent directory - and of course you can still use them that way if you wish.

4) You can also set the wildstar flag of AmigaDOS. Whenever this flag is set pattern "*" can be used instead of "#?".

Of course, nothing comes for free. Once UnixDirs3 is active you can't use names like "." or ".." for files/directories/links/etc. Multiparent option also prevents you using any name containing only periods. Trying to be bullheaded will cause the operation you were about to do failing with DOS error 204 - "Directory not found". Commodities Exchange can be used to temporarily toggle the UnixDirs3 off in case you still need to access an object with "illegal" name.

So that's for the brief summary. Now here comes even more brief one:

Things why to use UnixDirs3:

· It has been optimized for OS3, takes full advantage of it and - needs it. All OS3 DOS funtions that handle paths will be patched including file notification.

· Real cute multiparent option.

· One utility more to toggle DOS wildstar mode on/off.

· Commodity or no commodity. Use it how you prefer.

· Adjustable buffer sizes in case you are using huge path names.

· Does not fragment memory thanks to smart use of memory pools.

· When compared to UnixDirsII this version is faster (programmed in assembly instead of C), smaller and consumes less memory (with default settings).

· Again, the following problems of UnixDirsII fixed: no longer ejects filenames ending with two periods and doesn't crash with paths/patterns larger than 512 bytes.

Things why not to use UnixDirs3:

- You want to keep your Amiga system pure without any foreign *nix influence (good idea).
- You don't like programs safely patching your system libraries.
- You have a hard drive full of files with only periods in their names.
- You think that there are better ways to use 5 500 bytes of memory. (Or 2 500, if you are really tight of it.)
- Your horoscope adviced you so.

1.3 UnixDirs3 - Usage

```
                ---------------------------------------------------------------------------
                                Usage
---------------------------------------------------------------------------
```

First of all, UnixDirs3 can be started by double-clicking its icon or by typing its name right after the shell prompt. In the latter case, there is no need to "run" the program. To quit the program you can use one of the two starting methods (UnixDirs3 is clever enough to quit itself when started again), or in case you are executing UnixDirs3 without NOCX tooltype/argument (see below) you can also use commodities Exchange to quit the program or any third party utility to send a "CTRL-C" signal to UnixDirs3 commodity.

UnixDirs3 understands six different Workbench tooltypes and shell arguments. As they appear alike and there are no differences in their usage, I'll describe both arguments and tooltypes in this very same chapter.

The recognized options are:

```
XYZZY        <- argument/tooltype name (in case you are not quite sure, this
                really is an example, not a legal option)
 (switch)    <- type of argument - for switches the use of argument/tooltype
                will invert the default value (as all default values are off,
                they are thus toggled on), for numeric options you can see
                the legal range here.
 (off)       <- default value


QUIET        Normally (without this option) UnixDirs3 gives you a brief
             message everytime it's started or quited. This can be useful
 (switch)    when you are testing these arguments/tooltypes and have
 (off)       forgotten which action (starting/quiting) you are actually
             doing. However, after you have decided to
                install
                UnixDirs3
             and placed it for example in the Wbstartup drawer of your boot
             disk it is probably much more comfortable to ask it to keep
             its mouth shut. Thus this option. Note that errors and
             warnings will be reported no matter how this option is used.


NOCX         The usual thing is to start UnixDirs3 as a commodity. This way
             you can for example use commodities Exchange to temporarily
 (switch)    switch off the path manipulation so that you can access files
 (off)       with "illegal" names containing only periods (if multiparent is
             not specified then only "." and ".." will be illegal). Before
             any program can be made as a commodity, the commodities library
             has to be loaded from the disk. In case you are using UnixDirs3
             in the environment where it would be the one and only commodity
             in the system (not very likely) this option will perhaps save
             you some expensive bytes. Even if there actually are other
             commodities the use of this option will save about 3 000 bytes.
             Also note that quiting the program will give you this three
             kilobytes more free memory only in case this option wasn't
             specified at the moment the program was started. If it was,
             quiting the program will have no effect on the amount of free
             memory (see
               theory~of~operation
                for more details).


WILDSTAR     This option will set the corresponding flag in the DOS library
             base. When specified, it's possible to use an asterisk "*"
 (switch)    instead of "#?" pattern (match everything). The reason this
 (off)       isn't as default in the Amiga is that there is another use for
             the asterisk too. When used alone (like "RAM:*") the character
             refers to the particular console, it has nothing to do with
             filenames. The workaround would be to use "RAM:**" instead. If
             this option is not specified, then no change in the state of
             the flag will occur. However, if it was, the flag will be set
             and cleared again when the program is being quited. Using
             commodities Exchange to disable/enable the program while it's
             running will also have effect on the state of the wildstar flag
             if this option is being used.


MULTIPARENT  (My favourite one.) This option enables you to use more than two
             subsequent periods in order to refer to directories more than
 (switch)    one level upwards. So now you can use "..." instead of "../..",
 (off)       "...." instead of "../../.." and so on. One of the most common
             situations where this is used occurs when you are changing
             (CDing) your current directory with more than one step upwards.
             Thanks to OS3+ console device all you have to do is type (for
             example) "..." and there you go.


To really understand the last two options you should read the

               theory~of~operation
                chapter. People having headaches when it gets too
technical can forget them both. The default values are pretty good - and well,
```

even if the would not, nothing can seriously go wrong. So stop worrying and have fun with UnixDirs3!

```
BUFSIZE       This option sets the size for the buffer UnixDirs3 uses when
              manipulating the path string. If the value is too small (that
 (64-65536)   is, the buffer becomes full) the original string will be used.
 (512)        So if you sometimes notice UnixDirs3 not being able to correctly
              handle paths it should do, then please check this option.
              Perhaps the value should be a bit higher. See also the
                 note
                             below about buffers.


BUFNUM        Use this to adjust the amount of buffers UnixDirs allocates
              beforehand. This option is not so important after all, as new
 (1-256)      set of buffers will be allocated if there are no free ones
 (2)          available at a time they are needed. The reason for such a
              behaviour is to prevent memory fragmentation since there is no
              longer need to allocate and deallocate memory everytime buffers
              are being used. In normal circumstances the default sets of two
              buffers will be adequate. Don’t miss the
                 note
                  below.
```

Note about buffers: One set of buffers will be allocated at the first time UnixDirs3 is started and initialized. These buffers will not be freed until successfully allocating newer ones - not even when you quit the program (they will be used again when restarting UnixDirs3). Only when either - or both - of the BUFSIZE and BUFNUM parameters are being changed, the new allocation will occur and the old buffers are then naturally deallocated.

However, if there are no available buffers anytime when needed a new set of buffers will be allocated temporarily. This new set gets freed as soon as it is no longer needed.

To replace the old buffers requires that no one is using them at the moment. If there are any open file requesters etc. this can’t be done. Note also that this doesn’t prevent UnixDirs3 from being started, you just can’t override the previous buffer settings.

*   *   *

Here you can find additional
examples
.

1.4 UnixDirs3 - Installation

```
---------------------------------------------------------------------------
                              Installation
---------------------------------------------------------------------------
```

UnixDirs3 does not need any extra files, so you can just copy the program wherever you want to. Shell users may delete the icon file (UnixDirs3.info).

If you find the program appropriate in an everyday use you can place it into the WBStartup drawer of your system disk to get it started always when you boot up your Amiga.

For the shell users there is no need to "run" UnixDirs3. The program can be started anywhere in the system startup scripts.

Note however, that different versions of UnixDirs3 are not compatible. If you encounter a newer version of this program and want to see what's it for, you'll have to ensure that there are no other versions started previously. Depending on the installation method you'll have to either remove UnixDirs3 from the WbStartup drawer or from the startup scripts and reset your computer. Quiting the previously started version will not be enough!

1.5 UnixDirs3 - Theory of Operation

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

                          Theory of Operation
---------------------------------------------------------------------------
```

UnixDirs3 does its magic by rerouting some of the DOS library functions. In other words, all functions that handle paths will be patched. These include:

```
  AssignLate()
  AssignPath()
  CreateDir()
  DeleteFile()
  LoadSeg()
  Lock()
  MakeLink()
  MatchFirst()
  NewLoadSeg()
  Open()
  ReadLink()
  Rename()
  SetComment()
  SetFileDate()
  SetOwner()
  SetProtection()
  StartNotify()
```

Next time (after starting UnixDirs3, naturally) any of these functions is being used the execution first jumps into a newly added subroutine before the actual function. As you might have guessed, this subroutine resides in UnixDirs3.

There is one problem with patching. Adding a patch is easy, but removing one is never safe. It can be made almost 100% sure, but not fully. Never. As the functions UnixDirs3 patches are fairly important, I have decided not to remove patches at all. When the program is started first time, all sort of initialization occurs and the patches are installed. When you quit the program, only a flag is set to indicate that we are no longer active. And because the main program is rather small it is stored right there with the patches as well. This results in two things. One: once UnixDirs is started you have lost some kilobytes of memory forever (until next reset). Two: you cannot use different versions of UnixDirs3 after you have started the one you have started. These are not big problems as you probably have only one version of UnixDirs3, the memory loss is really negligible and no additional memory is allocated when you restart the program (old memory will be reused).

The subroutine takes care of substituting "." character at the beginning of the path into your current directory and it also handles multiple periods used to mean parent directory. If the current directory symbol (single period) is found in the middle of the path, the execution is stopped and returned with DOS error code 204 (directory not found).

Since the original DOS functions have no rights to alter the strings passed in (filenames, paths, etc.) no grants exist either for UnixDirs3. But the paths still need to be modified! This can only be done in one way. UnixDirs3 has to use additional buffer to copy the altered string into. As the use of buffers is fairly great (functions patched are very common ones), using the standard allocation method would likely mean memory fragmentation. This is why UnixDirs3 uses new feature of OS3+, namely memory pools. For the rest of the system one "puddle" of the pool looks like any allocated memory area. For the program using the particular pool it looks like just using the conventional method of memory allocation as long as there is room in the puddle being used to fulfill the allocation requests (one puddle can hold multiple allocations). And when there no longer isn’t any room left, new puddle (that is, new standard memory area) is allocated automatically - so nothing will be actually different. The puddle will be released back to other system’s use when all its allocations are freed (again, automatically). The idea behind all this is that there happens no real memory allocations as long as there is room in the puddles already in use - thus no fragmentation.

If you already are familiar with UnixDirs3
tooltypes/arguments
, you may
remember two options, namely bufnum and bufsize. From programmer’s point of view, a constant called puddlesize needed in initialization is calculated by multiplying these two values (or defaults, if they are missing). But that’s not enough. To make sure that a buffer always exist, the calculated size is added by four. This four bytes is then allocated instantly to ensure existence of a puddle.

Although DOS functions may work slower than most of the others in the system they are still fast enough to say that the buffers are only needed for a very short period of time. Again, there are exceptions. For example, if you try to access an invalid device a requester will appear (you know, the one saying "Please insert volume Foo: into any device; Retry - Cancel"). The function will not return until the requester has been closed. Meanwhile, the buffers can’t be deallocated (they are in use) and hence it’s not possible to change buffer settings.

Unlike the other features in UnicDirs3, the wildstar option is not a patch of any kind. There is already a flag in DosBase to indicate whether an asterisk is to be used as a wildcard or not. All there is to do is set or clear the state of this flag. Note especially, that leaving this option undetermined means no change in the current state of the flag. It does not mean clearing the flag.

1.6 UnicDirs3 - Examples

```
----------------------------------------------------------------------------
                                Examples
----------------------------------------------------------------------------
```

UnixDirs3 is a shell enchancer and so all examples are for the shell user as well. I have placed different type of examples right after the corresponding tooltype/argument. If you want to get all UnixDirs3 can give, then select following options: MULTIPARENT, WILDSTAR and possibly QUIET.

Well, here are the examples at last (you are not supposed to write the ">" character at the start of each exemplar line).

WILDSTAR

You may have activated this already (it's actually a feature of AmigaOS, not UnixDirs3) with some other system enhancer. If so, you don't have to activate it twice, once is enough.

To show all files in C: directory that have "T" (or "t") in their name:

```
> List C:*t*
```

To copy all files from your current directory to RAM: drive (note that using only one asterisk won't work):

```
> Copy ** RAM:
```

Oh no, so you tried using only one asterisk despite of all warnings... Your shell seems to be locked now, doesn't it? It surely does. Yep, nice thing. Well, since there seems to be nothing you could do, try writing something. Like "Hello, world" if your head feels surprisingly empty. Now, are we finally getting somewhere? You are supposed to press END-OF-FILE character right now. That's CTRL-\, usually. So try it. That should help.

If you had stated "Copy * Ram:Novel" you would now had the text you wrote in a file. As you can see, there exist an internal line editor in your console!

NO OPTIONS

To check what's inside a file "TheThing" in the parent directory of your parent directory:

```
> Type ../../TheThing
```

To copy the very same file to your current directory:

```
> Copy ../../TheThing .
```

To change your current directory to the parent directory of SOMEWHERE:

```
> SOMEWHERE:..
```

MULTIPARENT

To change your current directory three levels upwards (towards root):

```
> ....
```

Now let's do two of the tricks above again (the ones in NO OPTIONS). To check what's inside a file "TheThing" in the parent directory of your parent directory:

```
> Type .../TheThing
```

To copy the very same file to your current directory:

```
> Copy .../TheThing .
```

1.7 UnixDirs3 - Bugs

```
                ------------------------------------------------------------------------------
                             The Bug Page
----------------------------------------------------------------------------
```

There are, naturally, no bugs or errors in UnixDirs3. It's the other programs you are running in case some troubles arise ;)

Well, seriously, there are two things I should mention:

First, UnixDirs3 does the path manipulation in a system friendly way. So programs using their own routines for the job will work just like they did earlier. One common example is archiving program LhA. There isn't much to be done in cases like that.

I've also encountered at least one program that doesn't handle paths very sophistically (again, who knows, maybe it thinks being a smart one). Instead of just using the path as it is, it splits the path first into its components. With this kind of programs it's possible to use paths like './././././' without any DOS errors. However, in this particular case the path mentioned caused a nice crash (so the program wasn't so smart after all).

And naturally, in the very extraordinarily unbeliable situation there seems to be something wrong in the UnixDirs3, I'm the one to blame. Feel free to mail

me
and ask about corrected version.

1.8 UnixDirs3 - History

```
----------------------------------------------------------------------------
                                History
----------------------------------------------------------------------------


v1.0    the first published version

v1.1    a place of one library call changed (not likely, but still potential
        bug)

v1.2    ooops, using commodities exchange to quit the program could cause
        nasty things to happen. fixed

v1.3    oh no, this is becoming a habit. another bug found by Craig Hughes.
        apparently the ReadLink() patch has never worked as it trashed
        registers d4 and d5

v1.4    unixdirs3 no longer patches GetDeviceProc() function as this caused
        problems with programs using ixemul.library (they didn't recognize
        their current directory). i do also think that this particular patch
        wasn't really necessary

        From now on, UnixDirs3 is FREEWARE

v1.5    while reading The Amiga Guru Book written by Ralph Babel i found
        out that once upon a time (during os 1.3) some programs assumed dos
        functions return their results also in register d1 as that was the
        case at that time. an extra compability kludge was introduced in os
        2.04 because of this fact. these bad programs will now work with
        unixdirs3, too. furthermore, the patches are slight faster



----------------------------------------------------------------------------
                                Future
----------------------------------------------------------------------------


Shareware versions of UnixDirs3 (V1.0 - V1.3) were available about half an
year and I got no registrations. I understand that there no longer is any
interest left for paying shareware fees for simply system enhancers like this
program. I do finally release UnixDirs3 as freeware, but will there be any
versions after V1.5? Comments, please!
```

1.9 UnixDirs3 - Legalities

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

                              Legal Stuff
----------------------------------------------------------------------------
```

UnixDirs3 is freeware!

You are allowed to use UnixDirs3 free of charges providing the conditions below are not violated.

Copyright

UnixDirs3 is written by Timo~Kaikumaa. No parts of this program may be altered by any means (this includes editing, reprogramming, crunching, resourceing etc.), except archiving.

Disclaimer

You are using this program at you own risk! Under no circumstances will the author be liable for any direct or indirect damage or data loss ↩ resulting from the use or misuse of this software or the documents.

Distribution

UnixDirs3 can be freely copied as long as it is distributed with all the files in this package within. Only a nominal fee for costs of media may be accepted for distributing this piece of software. CD manufactures are specifically granted the right to include this program on CD collections, as long as they are for the Public Domain.

1.10 UnixDirs3 - Author

```
----------------------------------------------------------------------------
                              The Author
----------------------------------------------------------------------------


                             Timo Kaikumaa

                           Atanväylä 14 C 12
                             33580 Tampere

                                Finland


                        E-mail:  timok@cs.tut.fi

                  Home page (currently in Finnish only):
                       http://www.cs.tut.fi/~timok
```

1.11 Index

Index of database 00611d88-0

Documents

```
theory~of~operation

Timo~Kaikumaa

tooltypes/arguments
```