

ch_nfsc

COLLABORATORS

	<i>TITLE :</i> ch_nfsc		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		December 7, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	ch_nfsc	1
1.1	ch_nfsc.guide	1
1.2	ch_nfsc.guide/COPYING	1
1.3	ch_nfsc.guide/DISCLAIMER	2
1.4	ch_nfsc.guide/Limitations	3
1.5	ch_nfsc.guide/Overview	4
1.6	ch_nfsc.guide/Requirements	5
1.7	ch_nfsc.guide/Installation	5
1.8	ch_nfsc.guide/ch_nfsmount	6
1.9	ch_nfsc.guide/ch_nfsc	6
1.10	ch_nfsc.guide/Parameters	7
1.11	ch_nfsc.guide/Common	8
1.12	ch_nfsc.guide/Less Common	8
1.13	ch_nfsc.guide/Special	9
1.14	ch_nfsc.guide/Tuning	10
1.15	ch_nfsc.guide/ARexxPort	10
1.16	ch_nfsc.guide/Debugging	11
1.17	ch_nfsc.guide/ch_nfsctl	11
1.18	ch_nfsc.guide/Examples	14
1.19	ch_nfsc.guide/FAQ	15
1.20	ch_nfsc.guide/Reporting Bugs	17
1.21	ch_nfsc.guide/Implementat. Notes	17
1.22	ch_nfsc.guide/Thanks	19
1.23	ch_nfsc.guide/History	20
1.24	ch_nfsc.guide/Concept Index	22

Chapter 1

ch_nfsc

1.1 ch_nfsc.guide

Preface

This document describes the NFS client 'ch_nfsc' Version 1.01BETA and related commands. It gives some general information and describes how to install and use it.

Some legal stuff. * Read this before use *.

COPYING	Copyright stuff.
DISCLAIMER	Use at your own risk.
Limitations	Known Bugs, problems and limitations.

In medias res.

Overview	Introductory information.
Requirements	What's required?
Installation	Short overview about installation.
ch_nfsmount	Usage information about 'ch_nfsmount'.
ch_nfsc	Information about 'ch_nfsc' arguments.
ch_nfscctl	Usage information about 'ch_nfscctl'.
Examples	Some example mounts.
FAQ	Questions and answers.
Reporting Bugs	How to reach the author.
Implementat. Notes	Information about implementation details.
Thanks	Many people helped, here's their hall of fame.
History	A look back.
Concept Index	Direct way do various concepts.

1.2 ch_nfsc.guide/COPYING

Ch_nfs client

COPYING

=====

All files in this package are

Copyright (C) 1993, 1994 Carsten Heyl
All rights reserved.

You may freely copy this archive as long as it's contents are left
unmodified.

1.3 ch_nfsc.guide/DISCLAIMER

Information about warranty

=====

THIS CODE IS PROVIDED "AS IS" WITH NO WARRANTIES OF ANY KIND,
EITHER EXPRESSED OR IMPLIED. THE ENTIRE RISK OF USING THIS
CODE, PROGRAM OR INFORMATION IS ASSUMED BY THE USER. IN NO
EVENT WILL THE AUTHOR BE LIABLE FOR ANY DAMAGES, DIRECT,
INDIRECT, INCIDENTAL, SPECIAL OR CONSEQUENTIAL, RESULTING
FROM USING THIS CODE, PROGRAM OR INFORMATION, EVEN IF HE
(THE AUTHOR) HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGES.

This program uses code from the Sun RPC 4.0 distribution containing the
following note:

Sun RPC is a product of Sun Microsystems, Inc. and is provided for
unrestricted use provided that this legend is included on all tape
media and as a part of the software program in whole or part. Users
may copy or modify Sun RPC without charge, but are not authorized
to license or distribute it to anyone else except as part of a product or
program developed by the user.

SUN RPC IS PROVIDED AS IS WITH NO WARRANTIES OF ANY KIND INCLUDING THE
WARRANTIES OF DESIGN, MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE, OR ARISING FROM A COURSE OF DEALING, USAGE OR TRADE PRACTICE.

Sun RPC is provided with no support and without any obligation on the
part of Sun Microsystems, Inc. to assist in its use, correction,
modification or enhancement.

SUN MICROSYSTEMS, INC. SHALL HAVE NO LIABILITY WITH RESPECT TO THE
INFRINGEMENT OF COPYRIGHTS, TRADE SECRETS OR ANY PATENTS BY SUN RPC
OR ANY PART THEREOF.

In no event will Sun Microsystems, Inc. be liable for any lost revenue
or profits or other special, indirect and consequential damages, even if
Sun has been advised of the possibility of such damages.

Sun Microsystems, Inc.
2550 Garcia Avenue

Mountain View, California 94043

1.4 ch_nfsc.guide/Limitations

Known limitations, problems and bugs

=====

Limitations

1. Unix gids/uids > 0xFFFF will be mapped to <nobody> (0) on Amiga.
2. Filename parsing is a compromise between amiga and unix path usage.
Examples:
 - 'dir1/../dir2' becomes 'dir1'
 - 'dir1//dir2' becomes 'dir1'
 - 'dir1//../dir2' becomes ''
 - 'dir1./dir2' becomes 'dir1/dir2'

This makes using some unix symbolic links possible but may lead to confusion if mixed syntax is used.

3. The following DOS packets (actions) for file handlers are not supported by 'ch_nfsc':
 - 'LOCK_RECORD'
 - 'FREE_RECORD'
 - This can't be implemented with NFS alone. Possibly it can be implemented using the Lock Manager protocol via RPC. If someone is able to submit me any material about this protocol I may be able to support these.
 - 'ADD_NOTIFY'
 - 'REMOVE_NOTIFY'
 - Impossible via NFS if you're not the only user (correct me).
 - 'SERIALIZE_DISK'
 - Of any use for NFS?

Problems

1. Filename and directory parsing is done without regard to soft links. That means 'a/b/c/d///' will always give 'a/b' even if 'c' is a symbolic link pointing to another dir or even to file. The conversion will be done without even touching 'c' or 'd'.
2. Expect confusion when using '.' and '..' as normal filenames on the nfs handler.

Known bugs

1. Version 0.001 was accused of crashing the 'rpc.mountd' (the mount daemon) of a SGI running SVR4 and messing up the '/etc/xtab' file.

I didn't find a reason for that and have no information about later versions of 'ch_nfsc'.

For related information, see Implementat. Notes.

1.5 ch_nfsc.guide/Overview

General introductory information

=====

This is a NFS client implementing NFS Version 2. You should be able to remote mount any partition exported by a Version 2 NFS server. 'AmiTCP' >= 3.0 is required to run this software.

This program or a previous version has been run on various kinds of machines ranging from A500, A1000, A2000, A3000 to A4000.

The main machines to test were:

- Machine/Processor, MHz/OS Version/AmiTCP Version/Link Hardware
- A3000/68040, 35 MHz 68030, 16 MHz/2.04/3.0/Clone NE2000 Ethernetcard via GoldenGate II bridgecard
- A3000/68030, 25 MHz/2.1/3.0/Clone NE2000 Ethernetcard via Crosslink bridgecard

Tested NFS servers are:

- Interface(s)/Machine/OS
- Ethernet/Sparc clone running SunOS 4.1.1 NFS Server
- CSLIP, 14.4, .../Sun/SunOS 4.1.3 (CD-ROM)
- CSLIP, 14.4k, .../wuarchive:/archive
- CSLIP, 14.4k, .../Sun/SunOS
- Ethernet/386/Linux 0.99.14s
- Ethernet/Sparc-Station 2/Solaris 2.3
- ???/PC-Clone/DOS, SOSS 3.2
- Ethernet/486, 33 MHz/UHC SysV.4
- Ethernet/486DX2-66/FreeBSD V1.0.2

- CSLIP V2.6, 14.4k/Sun Sparc 10/SunOS 4.1.3
- Ethernet/DEC/Ultrix

(If you have problems with a special configuration, please tell, see Reporting Bugs.)

1.6 ch_nfsc.guide/Requirements

Requirements

=====

Hardware

The client should run on any Amiga. A minimum of 300 KBytes should be available to run a single client. If you plan to use multiple mounts at once think about making 'ch_nfsc' resident, it should be pure. Around 110K is needed for stack (30000 Bytes) and code (ca. 80 KBytes). The stack size needed will be reduced in the future.

Software

'AmiTCP' >= 3.0 is required.

The NFS Server you want to mount must support NFS Version 2.0 which is currently most widely used.

1.7 ch_nfsc.guide/Installation

Short installation guide

=====

To use 'ch_nfsc' you have to take the following steps:

1. Copy the binaries ('ch_nfsc', 'ch_nfsmount' , 'ch_nfsclnt', ...) to a directory in your path, e.g. 'AmiTCP:bin'.
2. Create a file 'AmiTCP:db/ch_nfstab'. This file contains information about the NFS volumes to mount, their local names and additional options, for all possible options see Parameters. Each line describes a different volume to mount. Lines containing a '#' will be ignored from '#' to the end of the line. The general structure is

```

    <hostname>:<remote device> <local device> [<additional options>]
e.g.
    nfserver:/usr/users/ch nfs: user ch

```

For more examples see Examples.

3. Use 'ch_nfsmount' to actually mount the specified volume, e.g.

```
ch_nfsmount nfs:
```

For details see ch_nfsmount.

1.8 ch_nfsc.guide/ch_nfsmount

'ch_nfsmount' usage information

=====

'Ch_nfsmount' is used to actually mount a NFS partition on your amiga. It needs a file describing the machine to mount from ('host'), the directory/partition to mount ('remote device') and the name the partition is locally known as ('local device'). This file is normally 'AmiTCP:db/ch_nfstab' but may be specified on the commandline, for examples see Examples. The argument template of 'ch_nfsmount' is

```
DEVICE,ALL/S,LIST/S,VERBOSE/S,FROM/K,OPTIONS/F
```

Before you can mount a partition from a remote host, this host must export the partition. This is usually done with an entry in '/etc/exports'. If you don't trust this program you may export it read only.

'DEVICE'

The device to mount, this is the 'local device' as specified in the 'ch_nfstab' file.

'ALL/S'

Mount all devices listed in the 'ch_nfstab' file.

'LIST/S'

Do not mount a device but display the contents of the 'ch_nfstab' file.

'VERBOSE/S'

Display the actual parameters as given to 'ch_nfsc' when mounting a volume.

'FROM <alternative 'ch_nfstab' file>'

Use an alternative 'ch_nfstab' file.

'OPTIONS/F'

The rest of the line are legal options of 'ch_nfsc' to specify e.g. the 'UMASK' or 'USER'. for details see ch_nfsc.

1.9 ch_nfsc.guide/ch_nfsc

Information about 'ch_nfsc' arguments.

=====

This section describes the arguments which may be given to 'ch_nfsc' via 'ch_fstab' or 'ch_nfsmount'.

Parameters	Parameters overview.
Common	Common command line parameters.
Less common	Less common command line parameters.
Special	Special command line parameters.
Tuning	How to get more speed.
ARexxPort	The integrated ARexx port.
Debugging	How to help locating bugs.

1.10 ch_nfsc.guide/Parameters

Parameters overview

REMOTE_DEVICE/A	Host to mount from and device to mount.
LOCAL_DEVICE	Local device name.
USER/K	User for NFS requests.
UMASK	Octal umask, used when creating files.
MNT_USER/K	User when mounting the partition.
MR	
MAX_READSIZE	Maximum number of data bytes used in RPC 'read'.
MD	
MAX_READDIRSIZE	Maximum number of data bytes used in RPC 'readdir'.
MW	
MAX_WRITESIZE	Maximum number of data bytes used in RPC 'write'.
B=BUFFERS	Initial number of 512 Byte buffers, default is 32.
ACTO	
ATTRCACHE_TIMEOUT	Timeout in seconds of cached attributes.
DCTO	
DIRCACHE_TIMEOUT	Timeout in seconds of cached dir entries.
RPCTO	
RPC_TIMEOUT	Maximum time to wait for RPC completion.
NO_TIMEOUTREQ/S	Don't use timeout requester.
SM/S	
SLOW_MEDIUM/S	Indicate 'using a slow transfer medium'.
ASYNC_RPC/S	Activate experimental asynchronous RPC.

Debug version parameters

DEBUG/K/N	Set debug level.
SERDEBUG/K/N	Set debug level, use serial port.
PARDEBUG/K/N	Set debug level, use parallel port.

1.11 ch_nfsc.guide/Common

Common command line parameters

`'REMOTE_DEVICE/A'`

Host and device to mount, template is `'<hostname>:<devicename>'`,
 e.g. `'wuarchive.wustl.edu:/archive'`
 Algorithm used to split: `'<hostname><first ':'><devicename>'`
 No further changes are made to these strings, they are used as got
 from the split above.

(Note: for some servers you may need to use `'host:dev:'` or
`'host:dev:\.'`)

`'LOCAL_DEVICE'`

Name the remote partition is locally known as, e.g. `"wustl:"`.

`'USER/K'`

User used for later NFS requests. Default is the user `'nobody'`.
 The name used must be a valid user listed in `amitcp:db/passwd`.

(Note: no passwd checking is done, so it's easy to fake a user)

`'UMASK/K/N'`

Octal umask used for all NFS requests dealing with file/directory
 creation, this number is octal in standard unix notation. The
 template is `0<user mask><group mask><others mask>` where `<mask>`
 consists of flags for read, write and execute in this order. If
 the corresponding umask bit is set the same file mode bit will be
 reset, e.g.

`'0077'`

Remove read/write/execute permission for group and others
 (this is the default)

`'0022'`

Remove write permission for group and others

`'BUFFERS/K/N'`

Initial number of 512 Byte buffers, default is 32. The number of
 buffers may be changed using the command `'AddBuffers'`.

`'NO_TIMEOUTREQ/S'`

Don't use timeout requester when encountering a RPC timeout but
 return an immediate error.

1.12 ch_nfsc.guide/Less Common

Less common command line parameters

`'MNT_USER/K'`

User used when mounting the partition, default in unix notation:
 user root (0), group wheel (0) The name used must be a valid user
 listed in amitcp:db/passwd.

(Note: No passwd checking is done, so it's easy to fake a user.)

Debug versions only:

.....

``DEBUG/N/K'`

Set debug level n, this number controls what information will be
 written to a log file "t:nfsh.log_<Tasknumber>"

- level 0: write all information
- level 1: write warnings and errors
- level 2: write only errors
- level 3: write no log messages at all default is level 2

``SERDEBUG/K/N'`

Same as ``DEBUG'` but writes output to serial interface at 9600 baud
 if not caught.

``PARDEBUG/K/N'`

Same as ``DEBUG'` but writes output to parallel interface.

(Note: Only one of ``DEBUG'`, ``SERDEBUG'` or ``PARDEBUG'` may be given on
 startup.)

1.13 ch_nfsc.guide/Special

Special command line parameters (tuning and troubleshooting)

``MR=MAX_READSIZE/K/N'`

``MD=MAX_READDIRSIZE/K/N'`

``MW=MAX_WRITESIZE/K/N'`

Defaults: 8192 bytes, for information when to use this, see FAQ.

``ATTRCACHE_TIMEOUT/K/N'`

Maximum time to cache attributes (Default: 30 secs).

(Note: When mounting readonly devices on slow links (e.g. SLIP)
 you may set this to several minutes.)

``DIRCACHE_TIMEOUT/K/N'`

Maximum time to cache directory structure (Default: 60 secs).

(Note: When mounting readonly devices on slow links (e.g. SLIP)
 you may set this to several minutes.)

``SLOW_MEDIUM/S'`

Give the handler a hint that the medium used is a slow one. This

is currently only used to use a different memory distribution between various buffers. It does not make much sense to use this Option alone. It should be accompanied with one of the 2 above.

``RPC_TIMEOUT/K/N'`

Maximum time that must be passed before an RPC call is considered as "timed out" (default: 25 seconds).

(Note: Within this time the RPC call is retried every `RPC_TIMEOUT/5` seconds, on slow links (e.g. SLIP) you may set this to a higher value to reduce unnecessary retries.)

``ASYNC_RPC/S'`

Activate experimental asynchronous RPC code. This should give some speed on some configurations, this will only have impact on reads or writes.

1.14 ch_nfsc.guide/Tuning

How to improve speed

When designing the NFS Client I had fast links in mind (ethernet, ...) but a high degree of people seem to try to use it on slow links like SLIP on direct links and even over modems. Directory walks and listings may be very slow on these configurations. To get faster response time for successive directory accesses you may try to increase `'ATTRCACHE_TIMEOUT'` and `'DIRCACHE_TIMEOUT'` and activate `'SLOW_MEDIUM'`. They default to 30 resp. 60 seconds which may not be sufficient for modem links see Special :-). To increase cache size use `'addbuffers'`.

Currently the only way to flush these caches is to reduce cache memory via `'addbuffers'`. Beware that you may get old information when setting these values too high and other NFS Clients change files/directories on the same server you use or you may run low on memory!

If you mount a read-only volume (e.g. a CD-ROM) it should be possible to set these values to several minutes without problems. They may be set on startup of `'ch_nfsc'`, see Special, or via `'ch_nfscctl'`.

Another way one can try to improve read/write speed by using asynchronous RPC which may be enabled with `ASYNC_RPC` on startup of `'ch_nfsc'`, see Special, or via `'ch_nfscctl'`.

Please tell me about your experiences! see Reporting Bugs.

1.15 ch_nfsc.guide/ARexxPort

The integrated ARexx port.

'Ch_nfsc' has an integrated ARexx port which may be used to set or query miscellaneous parameters. The port name is 'CH_NFSC' followed by an unique number. All commands have the form 'SET <something> <value>' or 'GET <something>'. The command 'GET COMMANDS' may be used to get a list of all commands and whether they support get or set. For examples see source code of 'ch_nfscctl.rexx'.

1.16 ch_nfsc.guide/Debugging

How to help locating bugs

'Ch_nfsc' is able to log all its actions into a file 't:nfsh.log_<tasknumber>'.

This behaviour will be activated by one of the following:

- supplying 'DEBUG 0' on the commandline when starting 'ch_nfsc'
- running 'ch_nfscctl device: set_debug 0' from the CLI

This will make all device accesses slow because a huge amount of data will be written to the log file.

Repeat your actions to reproduce the bug and make a copy of the log file after that. If you want to access the log file you need to shut off debugging so that 'ch_nfsc' closes the log file. This can be done with

```
ch_nfscctl device: set_debug 3
```

After that you will be able to access the log file in t:.

(For alternatives See ch_nfscctl.)

1.17 ch_nfsc.guide/ch_nfscctl

'ch_nfscctl' usage information
=====

'Ch_nfscctl' is used to set/query various parameters of a running NFS client 'ch_nfsc'. Since Version 0.007 it is an ARexx script. The communication is done via ARexx with port CH_NFSC<number>.

The usage template in general is:

```
ch_nfscctl <devicename> <arguments>
```

Currently supported commands: (Run 'ch_nfscctl' without arguments for usage message.)

```
'GET_ALL'
```

Display all actual settings.

```
'DIE'
```

Preliminary support for removing the handler from memory. The handler will refuse to remove itself from memory if open locks or file handles exist. In this case you will get a 'object in use' error.

Warning: It's possible to crash your machine with this command.

```
'FLUSH_DEBUG'
```

This command will force the handler to flush all buffered debug data to the log file.

```
'SET_DEBUG <debug level>'
```

Set new debug level, controls what to write into log file, possible values are

- level 0: write all information
- level 1: write warnings and errors
- level 2: write only errors
- level 3: write no log messages at all

```
'GET_DEBUG'
```

Display current debug level.

```
'SET_UMASK <new octal umask>'
```

Set umask to new value, e.g. 'SET_UMASK 007' (for description of format see unix book or see ch_nfsc).

```
'GET_UMASK'
```

Get current setting of umask.

```
'SET_USER'
```

Set effective user for NFS accesses to new value, e.g. 'SET_USER root'.

(Note: No password checking is done, so it's easy to fake a user.)

```
'GET_USER'
```

Get information about current effective user for NFS accesses.

```
'SET_MAX_READSIZE'
```

```
'SET_MR'
```

```
'GET_MAX_READSIZE'
```

```
'GET_MR'
```

Set/get maximum number of bytes requested from the NFS server in one 'read' call. Possible values are from 256 to 8192, the default

value is 8192.

``SET_MAX_READDIRSIZE'`

``SET_MD'`

``GET_MAX_READDIRSIZE'`

``GET_MD'`

Set/get maximum number of bytes requested from the NFS server in one `'readdir'` call. Possible values are from 256 to 8192, the default value is 8192.

``SET_MAX_WRITESIZE'`

``SET_MW'`

``GET_MAX_WRITESIZE'`

``GET_MW'`

Set/get maximum number of bytes written to the NFS server in one `'write'` call. Possible values are from 256 to 8192, the default value is 8192.

``SET_WRITEBUFFERLIMIT'`

``SET_WL'`

``GET_WRITEBUFFERLIMIT'`

``GET_WL'`

Set/get maximum number of bytes of a single `Write()` getting buffered. Set this to 0 to disable buffer for further writes.

``SET_ATTRCACHE_TIMEOUT'`

``SET_ACTO'`

``GET_ATTRCACHE_TIMEOUT'`

``GET_ACTO'`

Set/get maximum time in seconds a cached attribute will remain valid. The default value is 30 seconds.

``SET_DIRCACHE_TIMEOUT'`

``SET_DCTO'`

``GET_DIRCACHE_TIMEOUT'`

``GET_DCTO'`

Set/get maximum time in seconds a cached directory entry will remain valid. The default value is 60 seconds.

``SET_RPC_TIMEOUT'`

``SET_RPCTO'`

``GET_RPC_TIMEOUT'`

``GET_RPCTO'`

Set/get maximum time in seconds that must be passed before a RPC call is considered as "timed out", The default value is 25 seconds. (On slow links (e.g. SLIP) you may set this to a higher value to reduce unnecessary timeouts and retries.)

(Note: Within this time the RPC call is retried every `RPC_TIMEOUT/5` seconds.)

``SET_TIMEOUTREQ <0 or 1>'`

``SET_TR <0 or 1>'`

``GET_TIMEOUTREQ'`

``GET_TR'`

Disable/enable timeout requester flag resp. query current setting.

```
`SET_SLOW_MEDIUM <0 or 1>'
`SET_SM <0 or 1>'
`GET_SLOW_MEDIUM'
`GET_SM'
    Disable/enable slow medium flag resp. query current setting.
```

```
`SET_ASYNC_RPC <0 or 1>'
`SET_ARPC <0 or 1>'
`GET_ASYNC_RPC'
`GET_ARPC'
    Disable/enable asynchronous RPC resp. query current setting.
```

Examples

1. Change the effective user for NFS accesses to user 'ftp', the NFS partition is mounted as 'NFS:'.
`ch_nfscctl NFS: set_user ftp`
2. Query all settings.
`ch_nfscctl NFS: get_all`

1.18 ch_nfsc.guide/Examples

Some examples

=====

Example 'AmiTCP:db/ch_nfstab' file:

```
#
# REMOTE DEVICE, LOCAL DEVICE, OPTIONS/F
#

wuarchive.wustl.edu:/archive      wustl:
work-machine:/home                nfs-home: USER ch UMASK 0027
# Note: We assume that the archive below is read only.
wuarchive.wustl.edu:/archive      wustl-arc: ATTRCACHE_TIMEOUT 600 \
    DIRCACHE_TIMEOUT 600
```

(Note: The '\\' above indicates that the full entry must be written in a single line.)

Example 1

Mount the directory '/archive' of host "wuarchive.wustl.edu" as local partition "wustl:".

```
ch_nfsmount wustl:
```

Example 2

Mount the directory '/home' of host "work-machine" as local partition

"nfs-home:".

Do all NFS accesses (reading/writing/creating) as user 'ch' using the group specified for 'ch' in the amictcp 'passwd' file. Members of the same group as 'ch' should be able to read or execute but not write the files we create but others should not get anything.

```
ch_nfsmount nfs-home:
```

Example 3

Do the same as Example 1 above but here we assume that the archive is read only so that raising cache timeouts will not harm.

```
ch_nfsmount wustl-arc:
```

Example 4

Do the same as Example 3 but over a slow line, e.g. a modem. This also shows how to add extra arguments via command line.

```
ch_nfsmount wustl-arc: slow_medium buffers 500
```

(Note: NFS is not the best choice when reading files over a slow line. I did some work to get better performance through local caches but you should use 'ftp' or better 'ncftp' for file transfers if possible.)

1.19 ch_nfsc.guide/FAQ

Some questions and answers

=====

'Q.1'

Normal file and directory browsing is o.k. but some programs are really slow when accessing files via NFS.

'A.1'

Some programs do their IO (reads/writes) in very small pieces. The handler tries to buffer writes for files opened with MODE_NEWFILE and tries to read ahead but if that's not possible each separate read or write action will result in the synchronous call of the read/write procedure on the NFS server. This is certainly slow but don't blame me for poorly written software. You may also try to increase the buffer memory used with 'addbuffers'. One of the programs that don't behave well when used on a NFS mounted partition is SAS/C 6.3. It does a lot of it's write IO doing Seek()/Write() pairs for small data portions.

'Q.2'

Everything else works o.k. but I always/sometimes get RPC Timeouts (Error code 1005) when

- a. listing large directories

b. reading files > ca. 8K

c. writing files > ca. 8K

'A.2'

Try to set the respective maximum transfer size as a startup option see Special or via 'ch_nfscctl' see ch_nfscctl.

'ethernet:'

Start with 1000 bytes and test that and if that's o.k. try to get more closer to the maximum possible number. Please report what you get!

'On slip lines:'

Start with 800 bytes.

'Reason:'

Try to avoid IP fragmentation, because your server/router(s) may have problems with this and in case of the loss of one piece the whole (fragmented) IP packet has to be re-transmitted. Another reason may be your local ethernet card being too slow. Try to get the SANA II statistics from it and look for overruns

'Q.5.'

'C:Type' works for me but some programs (e.g. less 1.6z) report 'not a regular file' and don't show the file contents.

'A.5.'

This problem needs further observation. My guess: These programs may have problems with extra permissions (group/other) returned by Examine().

'Q.6'

When I start the NFS Client I get an error code without any explanation.

'A.6.'

Ch_nfsc should display an error requester before returning an error code. You may however suppress requesters by using a 'noreq' utility or setting the _NOREQ variable when using the 'csh'.

'Q.7'

I start 'ch_nfsc' with debug option but 'list' lists a length of 0.

'A.7'

Due to speed considerations a big buffer is used when writing debug information into a file. Use the 'FLUSH_DEBUG' option of 'ch_nfscctl' to flush buffer, see ch_nfscctl.

'Q.8'

When listing files or directories on NFS mounted volumes the Time is off a few hours.

'A.8'

Did you set the timezone environment variable 'TZ' to your correct timezone? But note that the value of 'TZ' is read when the NFS volume is mounted. So changing 'TZ' does not have impact on running NFS clients.

1.20 ch_nfsc.guide/Reporting Bugs

How to reach the author.

=====

If you find a bug please send me a detailed report so I will be able to understand it and possibly reconstruct or reproduce and fix it. If possible send me a copy of the log-file see Debugging.

```
*****  
* STOP *  
*****
```

Before you send a mail please make sure it contains more than "Hey, I tested you program. But sometimes it does not function."

Just take the time to re-read your mail and try to answer the following:

1. Would YOU be able to get any useful information from YOUR mail?
2. Does it contain enough information about the used hardware/software and enough details on the problem?

If you think you supplied sufficient information please send it to me.

My current email address is ch@irb.informatik.uni-dortmund.de. If that's no longer valid you may try to reach me via snail mail:

```
Carsten Heyl  
Unterer Markt 4  
49477 Ibbenbueren  
Germany
```

Feel free to add suggestions for improvements and the like ... I'm open to new ideas but can't make promises.

1.21 ch_nfsc.guide/Implementat. Notes

General information about this implementation

=====

General

First of all I hope you will find this program useful. The RPC library used is based on Sun's RPC 4.0 distribution, see DISCLAIMER. Prior to version 1.0 the RPC library used was ported by Jarno Rajahalme <Jarno.Rajahalme@hut.fi>, a member of the AmiTCP group, with some small changes by me. As of Version 1.0 of 'ch_nfsc' the RPC library as distributed with 'AmiTCP' 3.0 is used. The mount/nfs stubs were generated by my port of the rpcgen program of the Sun TI (Transport Independant) rpc distribution. That version of rpcgen should be included in the 'AmiTCP' 3.0 distribution.

Implementation assumptions

1. This NFS handler will only properly function with nfs servers which use '.' and '..' for directory linking or emulate that behaviour.
2. It is not legal to make a memory copy of a file lock for locks obtained from this handler (doing this is considered illegal (see 'AmigaDOS Manual') but nevertheless done by some programs).
3. The Amiga FS's are case insensitive but Unix Filesystems are case sensitive. The algorithm to find a file is currently as follows:
 - a. A lookup for exactly the given name is made.
 - b. If not succesful: A ReadDir is made on the given 'dir' and the first (caseinsensitive) match will be found (if any).

Note that this may be slow on large directories, so avoid different case if possible. The procedure above is only necessary when opening or listing files, it is *NOT* necessary for reads or writes on open files.
4. The nfs-server must support '..' as a name for parent directory ('nfsproc_lookup_2(<dir-handle>, "..")' must return parent dir).
5. 'Lookup' fileid and 'getattr' fileid must match.

User/group ids on my amiga?

As of AmigaOS 3.0 the amiga knows about user and group information for files. This includes information about extra protection bits for others and group and information about the owner (UID/GID) of a file. Since I could not find any public information about this I gathered all information from others' sources and from the 3.0 includes.

Here is the information I got. The amiga seems to use other defaults than unix so some remapping needs to be done:

```
'amiga'
      UID 0          - user "nobody"
      UID 1-65534    - normal user ids
      UID 65535     - user "root"
```

```
GID 0      - group "nobody"
GID 1-65534 - normal group ids
GID 65535  - group "wheel"
```

`unix'

```
uid 0      - user "root"
uid 1-65534 - normal user ids
uid 65535  - user "nobody" (-1)

uid 0      - group "wheel"
uid 1-65534 - normal group ids
uid 65535  - group "daemon" (nobody, -1)
```

So the obvious conversions will be applied when necessary.

(Note: User/group id's from 'AmiTCP:db/passwd' resp. 'AmiTCP:db/group' will *NOT* be converted.)

What gets cached/buffered and how

'Ch_nfsc' does a caching of

- file/directory attributes (normally valid for 30 secs, see Tuning.)
- Some directory structure information (normally valid for 60 secs, see Tuning.)
- 'Write()'s' less or equal than WRITEBUFFERLIMIT (default: 16K) are buffered to reduce net use and increase speed for programs doing writes in small pieces.
(Note: Only files opened with MODE_NEWFILE will be write buffered)
- Small reads will be satisfied from a read buffer if possible.

1.22 ch_nfsc.guide/Thanks

Thanks and acknowledgements

=====

Lot's of people did their best to test this program and to locate bugs but as always:

Bugs will still remain, use at your own risk!

First I have to thank the AmiTCP group for porting a public available TCP/IP stack to the amiga and for answering questions/locating bugs. Without their effort I wouldn't had even think about writing this software.

Thanks go to the following for their help in reporting and locating bugs and problems and sending information and suggestions: (I hope I didn't miss one.)

Stefan G. Berg, Kenneth Scott Bethke, Mike Bond, Joe Elliot, Michael D. Fischer, Joerg Cyril Hoehle, Holger Hofstaette, Arve Hjoennevaag,

Oliver Huf, Kim-Minh Kaplan, Lee S. Kilpatrick, Charles Macmillan, Rudolf Neuhaus, Clive Nicolson, Ross Niebergall, Stephen Norris, Timo Rossi, K. Raquel Sanborn, Barnon "Barry" Tigner, Christian Wolf

Thanks go to Randell Jesup and Chris Hooper for answering questions and sending information about special DOS packets.

Thanks go to Ralph Babel for his book 'The Amiga Guru Book' and the informative parts of his mails :-)

(Even if I don't like the index of that book it's very useful to get information not found anywhere else.)

Special thanks go to Marcus Kommnick for his help and patience in testing various hardware and software components sometimes with life spawns of few minutes :-) and for cross-reading the documentation.

1.23 ch_nfsc.guide/History

History

=====

'1.01BETA'

- * fixed bug when writing 0 bytes
- * fixed 2 bugs in buffer stuff (these might lead to data loss!)
- * recognizing timezone now
- * make shutting off ASYNC_RPC true

'1.0BETA'

- * AMITCP 3.0
- * async support
- * ch_nfsmount: fix fstab parsing bug, test for device/volume before mounting now
- * added ARexx port, added short parameters
- * some small fixes

'0.007'

- * internal version

'Version 0.006 (BETA)'

- * fixed wrong assertion if early mount failure
 - * re-activated "invalid argument" requester
 - * fixed time conversion bug
-

- * fixed ch_nfsmount bug if ALL option was used (may crash machine/process)
- * fixed freed memory reference if illegal passwd entry
- * fixed timer problem when mount failed

`Version 0.005 (BETA)'

- * fixed memory leaks in Lock(), dir_DelEntry(), CBuildFullName(), do_act_Parent()
- * fixed 2 subtle/serious memory bugs filename handling stuff
- * added file type check to READ_LINK
- * fixed delete/write mapping of SetProtection
- * located FComp problem, packet arguments were modified, fixed now
- * (rough) memory limit is now adjustable via AddBuffers
- * added preliminary die support (use at your own risk)
- * added SLOWMEDIUM hint
- * mapped Amiga "nobody"/"nogroup" to unix "-1"
- * amiga guide doc's

`Version 0.004 (BETA)'

- * special BETA release
- * some minor memory enhancements
- * added serial/parallel debug options
- * action CURRENT_VOLUME supported now
- * find_update: make error message if open on non-file ffs like

`Version 0.003 (BETA)'

- * third release, to BETA testers only
 - * added RPCTimeout parameter
 - * changed ch_nfscctl to ReadArgs()
 - * hopefully fixed assign-bug
 - * some speedup to dir cache lookup
 - * added read ahead buffer
 - * made error conversion context sensitive -> better error reporting
-

```
* Requester now shows programname instead of "Background CLI"

* main(): Added explicit stack size check on startup

* some documentation fixes

* added a timer for automatic buffer/cache flushes

`Version 0.002 (BETA)`
* second release, to BETA testers only

* added some support for slow lines (see Tuning)

* much more configuration parameters (too much?)

* Read(): avoid extra nfs call on EOF

* Lock(): fixed error reporting bug ("not found" was reported
    as "out of memory")

* return volumename on root lock now (thanks to Timo Rossi)

* fixed serious memory bug in FIND_OUTPUT if second open on
    exclusive          write is tried

* fixed path parsing bug (a leading "/" is now always found and
    considered illegal)

* fixed path parsing bug regarding "..", "/.." is now
    identical to amiga notation "/"

* improved detection of illegal FIB contents presented to
    ExNext()

* added write buffer for small write speedup

* added debug parameters for maximal read/write/readdir size

* minor changes/fixes

* fixed error reporting bug (most RPC errors were reported as
    "out of memory" in 0.001)

* added RPC Timeout requester

`Version 0.001 (BETA)`
* first public release
```

1.24 ch_nfsc.guide/Concept Index

Concept Index

ch_nfstab

Author

DISCLAIMER

Debugging

Installation

Questions and answers.

Thanks

Timezone

Tuning

Installation

Reporting Bugs

DISCLAIMER

Debugging

Installation

FAQ

Thanks

FAQ

Tuning
