Version 1.1 – August 19, 2020

ESC NFS Documentation

Table of Contents:

Engineering & Science Computing
University of Notre Dame

ESC NFS Overview

Engineering & Science Computing's implementation of NFS is designed to meet the needs of Linux users within the colleges of Engineering and Science. NFS is available in the computer clusters in Engineering classrooms and the Engineering library, as well as the remote virtual machines.

Local Installation

For personal Linux workstations that need direct connectivity to the ESC NFS system, please contact the ESC Help Desk for assistance in configuration (email: help@esc.nd.edu phone: 631-0101). ESC has also created NFS Client setup instructions if you wish to configure your workstation(s) yourselves. Below are the links:

- Ubuntu NFS Client Setup
- RHEL 7 NFS Client Setup
- Debian 9.5 NFS Client Setup
- Fedora 29 NFS Client Setup

Accessing NFS

The recommended way to access NFS is via SSH using the ESC Remote300 machines. This allows users to access the system from any standard SSH client, placing users in their home directory on the system. Users accessing NFS from different systems will find their data is the same whether they SSH into the remote hosts, via login on the Linux computer clusters, or from specified CSE remote hosts.

There are 8 remote machines, but you can connect to **remote300.helios.nd.edu** to connect to one of them at random. This acts as a load balancer and prevents an unfortunate side effect of users all trying to connect to the first few nodes, while the last few remain relatively unused.

You can connect using the following command:

```
ssh netid@remote300.helios.nd.edu
```

Once connected you will be placed in your home directory. You can locate your full home directory path using the print working directory command (pwd)

```
$ ssh jslaught@remote300.helios.nd.edu
********************************************************************************
********************************************************************************
**                                                                            **
**     This system is for academic computing (classes and lab clusters)       **
**               for Colleges of Engineering and Science only.                **
**                                                                            **
**             This system is not set up for research purposes.               **
** For research computing, please contact the Center of Research Computing.   **
**                                                                            **
**       Remote use of this system is not for running big processes.          **
**                                                                            **
**      See the University of Notre Dame Responsible Use of Information       **
**     Technology at https://oit.nd.edu/about-us/policies-and-standards/      **
**                                                                            **
********************************************************************************
********************************************************************************

       Please email help@esc.nd.edu for any questions or suggestions.
jslaught@remote300.helios.nd.edu's password:

Last login: Mon Jan 21 09:09:45 2019 from 172.0.0.7
$ pwd
/escnfs/home/jslaught
jslaught@remote307:~$
```

As you can see this placed me on the *remote307* server, in my home directory located at */escnfs/home/jslaught.*

Permissions

The security model for NFS is a restrictive model, allowing only the owner of a file access to it regardless of where the file is created. Keep this in mind as you create files in areas where an expectation of sharing or collaboration may exist. Further effort may be needed to ensure the proper people have access to these files. See Situations With Less-Restrictive Permissions for more information about how ESC handles these scenarios.

The default ownership of all new ***files*** in NFS use bitwise permissions of 600, like so:

	Read	Write	Execute
Owner	Yes	Yes	No
Group	No	No	No
Everyone	No	No	No

The default ownership of all new ***directories*** in NFS use the 8-bit permission of 700, like so:

	Read	Write	Execute
Owner	Yes	Yes	Yes
Group	No	No	No
Everyone	No	No	No

All users belong to the default group of **notre-dame**. If you allow permission to this group, you may end up allowing anyone in the NFS system to access your data. In this scenario, the group and everyone permissions act effectively the same.

Software Modules

NFS provides a common set of modules for loading software consistently within the Linux systems through SSH as well as in the classrooms.

To list available modules, you can type:

```
module avail
```

Currently we have 4 modules available for the following software packages, with the syntax for invoking them below:

- Advanced Design System 2019
- Intel Fortran Compiler 18.0
- MATLAB R2018b
- Mathematica 11.3

Advanced Design System

ads/2019

```
module load ads
```

Intel Fortran Compiler

intel/18.0

```
module load intel
```

MATLAB

matlab/R2018b

```
module load matlab
```

Mathematica

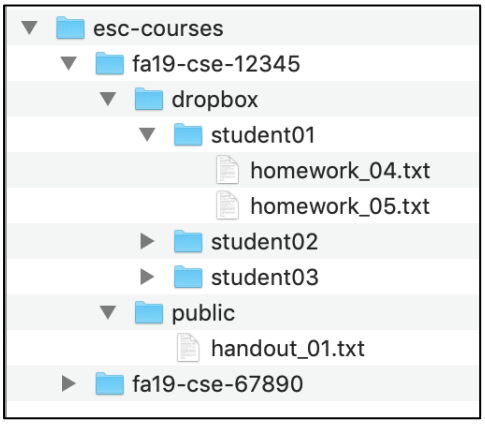
mathematica/11.3

```
module load mathematica
```

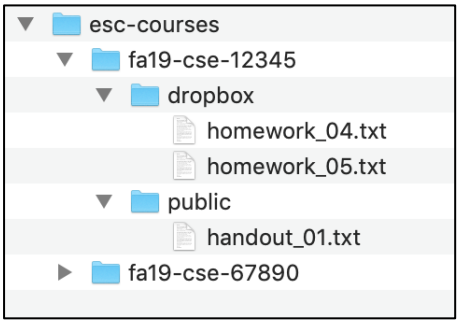
Considerations for Courseware

Instructors who require a Courseware-like environment within NFS can request a course folder. A symbolic link to it will be placed in users **~/esc-courses** directory for all instructors, TAs, and students.

For instructors and TAs, the course folder will contain the following structure, as shown here with sample data:



Students, however, will not see a list of other students. They will only get a link to their own contents:



These folders are designed to meet specific requested features of courseware, specifically places for students to place work, and a public folder to contain general course materials available to any student.

Situations With Less-Restrictive Permissions

Permissions differ for situations like Courseware due to the nature of sharing work with others. This difference is not inherent, but maintained by a script that runs on the server every quarter hour to ensure faculty and TAs always have full access to all folders and files in their courses. If a student, TA, or Instructor cannot access a file, the permissions will likely be corrected on the next quarter hour. If not, please contact ESC and we will look into your specific concern.

The script creates a course group (name is identical to the course id) that contains all instructors and TAs for a course, and applies a read/write/execute permission for that group recursively on everything in the course directory. This way, the persons leading the course will always have access to content placed there by either students or other TAs/instructors, no matter who the owner is set to or how it may have been placed there. Below is an example of the faculty view of a course folder, with a course group name of "fa19-cse-12345.01".

```
drwxrws--x  9 esc    fa19-cse-12345.01  119 Apr 12 21:14 .
drwxrwxr-x 22 root   esc-nfs-admins    4096 May 29 14:52 ..
drwx------  2 netid  fa19-cse-12345.01    6 Apr 12 22:19 admin
drwxrws--x 50 esc    fa19-cse-12345.01 4096 Apr 25 09:31 dropbox
drwxrwx---  2 netid  fa19-cse-12345.01 4096 Mar 20 15:29 internal_folder
drwxrwsr-x  6 esc    fa19-cse-12345.01  107 Mar 22 16:04 public
```

Once the script runs, permissions on the pre-populated folders will *effectively** be as follows:

- **dropbox** (Student full access)
  - student (owner) = read/write/execute
  - Instructors & TAs (group) = read/write/execute
  - Everyone = no permissions
- **public** (Read-only, all students)
  - student (owner) = read/execute
  - Instructors & TAs (group) = read/write/execute
  - Everyone = read/execute

* *Note*: The **dropbox** directory may appear to have greater permissions via the "ls" command, but will be restricted for each subdirectory via specific NFSv4 permissions. For more information about setting and reading NFSv4 permissions, see the Advanced Topics section.

Additional directories created in the course parent directory will never be accessible by students unless permissions are explicitly set.

To see the members of a course group run the following command:

```
ldapsearch -x -h addc11-prod.nd.edu -D <your netid>@ND.EDU -W -b "OU=Campus,DC=ND,DC=EDU"
  "sAMAccountName=<course id>" member
```

Advanced Topics

NFS Permissions

Basics of NFSv4 ACLs

NFS ACLs are split up using colons for each access control entry (ACE), with each ACE as follows:

ACE Type:

A	Allow
D	Deny

ACE Flags:

d	New directories will inherit the ACE
f	New files will inherit the ACE
n	ACE will apply to this item, but will not be inherited by subdirectories and new files
I	ACE will not apply to this item, but will be inherited by subdirectories and new files
g	This flag is used when the principal below is a group, not a user

ACE Principal:

Principals for users are email addresses, like netid@nd.edu, or group@nd.edu.

ACE Permissions:

r	read-data (files) / list-directory (directories)
w	write-data (files) / create-file (directories)
a	append-data (files) / create-subdirectory (directories)
x	execute (files) / change-directory (directories)
d	delete the file/directory
D	delete-child : remove a file or subdirectory from the given directory (directories only)
t	read the attributes of the file/directory
T	write the attribute of the file/directory
n	read the named attributes of the file/directory
N	write the named attributes of the file/directory
c	read the file/directory ACL
C	write the file/directory ACL
o	change ownership of the file/directory

Permissions Shorthand:

Alias	*Name*	*Expansion*
R	Read	rntcy
W	Write	watTNcCy (with D added to directory ACE's
X	Execute	xtcy

Commands:

-a acl_spec [index]	add ACL entries in acl_spec at index (DEFAULT: 1)
-x acl_spec | index	remove ACL entries or entry-at-index from ACL
-A file [index]	read ACL entries to add from file
-X file	read ACL entries to remove from file
-s acl_spec	set ACL to acl_spec (replaces existing ACL)
-S file	read ACL entries to set from file
-m from_ace to_ace	modify in-place: replace 'from_ace' with 'to_ace'

Options:

-R	recursive	Applies ACE to a directory's files and subdirectories
-L	logical	Used with -R, follows symbolic links
-P	physical	Used with -R, skips symbolic links

Example Permissions:

If the item your adding permission to is a directory, you may have to apply an ACE to the item and an ACE for managing inheritance.

Grant user recursive read/write/execute access to a folder:

```
nfs4_setfacl -R -a A:df:jslaught@nd.edu:RWX directoryname
```

Grant everyone recursive read/execute access:

```
nfs4_setfacl -R -a A:df:EVERYONE@:xtcy directoryname
```

Document History

06/05/2019	1.0	Initial release.
08/19/2020	1.1	Added an ldapsearch example for course group membership lookup