Introduction au système de fichiers réseau NFSv4

Philippe Latu
philippe.latu(at)inetdoc.net

https://www.inetdoc.net

Résumé

L'objectif de ce support de travaux pratiques est d'étudier le système de fichiers réseau NFS. Il illustre les accès en « mode fichier » à une unité de stockage réseau. Ce mode d'accès correspond à un stockage de type NAS (*Network Attached Storage*). Le document commence par l'étude du principe de fonctionnement des appels de fonctions RPC (*Remote Procedure Call*), puis se poursuit avec la configuration d'un serveur NFS qui exporte une arborescence de comptes utilisateurs. Côté client, les accès au système de fichiers réseau NFS sont étudiés selon deux modes distincts : le montage manuel, puis l'automontage.

Table des matières

1. Copyright et Licence 1
2. Topologie, scénario et plan d'adressage 2
3. Protocole NFSv4 3
    3.1. Architecture NFSv4.2 3
    3.2. Sécurité NFSv4.2 4
    3.3. Adoption NFSv4.2 et innovations 4
4. Configuration commune au client et au serveur NFS 4
    4.1. Gestion des appels RPC 5
    4.2. Gestion des paquets NFS 7
5. Configuration du serveur NFS 8
6. Configuration du client NFS 11
    6.1. Opérations manuelles de (montage|démontage) NFS 11
    6.2. Opérations automatisées de (montage|démontage) NFS 13
7. Gestion des droits sur le système de fichiers NFS 15
8. Documents de référence 16

1. Copyright et Licence

```
Copyright (c) 2000,2025 Philippe Latu.
Permission is granted to copy, distribute and/or modify this document under the
terms of the GNU Free Documentation License, Version 1.3 or any later version
published by the Free Software Foundation; with no Invariant Sections, no
Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included
in the section entitled "GNU Free Documentation License".
```

```
Copyright (c) 2000,2025 Philippe Latu.
Permission est accordée de copier, distribuer et/ou modifier ce document selon
les termes de la Licence de Documentation Libre GNU (GNU Free Documentation
License), version 1.3 ou toute version ultérieure publiée par la Free Software
Foundation ; sans Sections Invariables ; sans Texte de Première de Couverture,
et sans Texte de Quatrième de Couverture. Une copie de la présente Licence est
incluse dans la section intitulée « Licence de Documentation Libre GNU ».
```

Méta-information

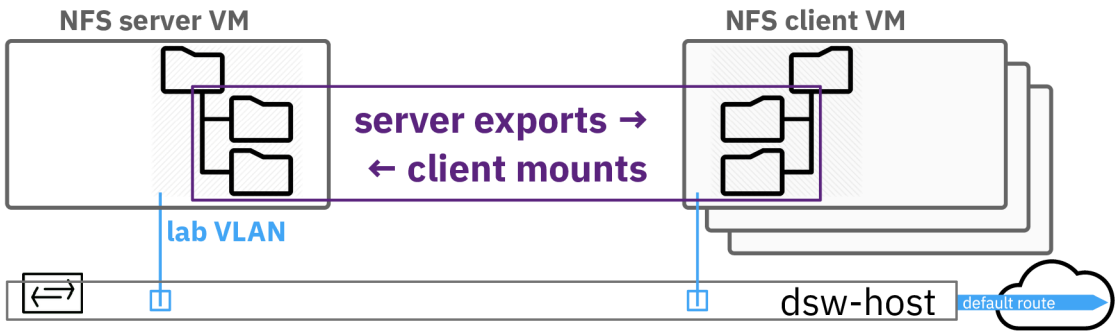
Ce document est écrit avec *DocBook* XML sur un système *Debian GNU/Linux*. Il est disponible en version imprimable au format PDF : `sysadm-net.nfs.pdf`.

2. Topologie, scénario et plan d'adressage

Topologie logique

Les manipulations présentées dans ce support utilisent un domaine de diffusion unique (VLAN) dans lequel on trouve au moins deux systèmes virtuels ou physiques avec deux rôles distincts.

- Le système *`serveur exporte`* une arborescence de son système de fichiers local vers les clients.
- Le ou les systèmes *`clients montent`* le système de fichiers réseau sur une arborescence locale.



Topologie logique - vue complète

Scénario

Les manipulations décrites dans ce document illustrent les fonctionnalités offertes par le protocole NFS. Le séquencement des opérations à réaliser lors de la séance de travaux pratiques est décrit dans le tableau ci-dessous. Après le traitement de la première partie commune, les deux systèmes jouent un rôle distinct. Le rôle client accède à l'arborescence partagée ou exportée par le rôle serveur.

Tableau 1. Attribution des rôles NFS

Client	Serveur
Identifiez du mécanisme des appels RPC. Installez et configurez les paquets communs.	
Identifiez des services disponibles sur le serveur. Créez un compte local sans répertoire utilisateur.	Installez le paquet spécifique au serveur et configurez le service en fonction de l'arborescence à exporter.
Validez l'accès au système de fichiers réseau avec capture de trafic.	
Installez le paquet spécifique et configurez le service d'automontage des répertoires utilisateurs.	

De nombreuses questions peuvent être traitées à l'aide du document de référence *Nfsv4 configuration* pour ces travaux pratiques. Il faut toutefois faire correspondre les configurations décrites dans ce document avec celles proposées par les paquets de la distribution Debian GNU/Linux.

Plan d'adressage

Partant de la topologie présentée ci-dessus, on utilise un plan d'adressage pour chacun des rôles iSCSI.

Le tableau ci-dessous correspond au plan d'adressage de la maquette qui a servi à traiter les questions des sections suivantes. Lors des séances de travaux pratiques, un plan d'adressage spécifique est fourni à chaque binôme d'étudiants. Il faut se référer au document *Infrastructure*.

Tableau 2. Plan d'adressage de la maquette « Introduction au système de fichiers réseau NFSv4 »

Rôle	VLAN	Adresses IP	Interface tap
Client NFS	101	`172.28.101.6/24` `2001:678:3fc:65:baad:caff:fefe:6/64`	6
Serveur NFS	101	`172.28.101.5/24`	5

Rôle	VLAN	Adresses IP	Interface tap
		`2001:678:3fc:65:baad:caff:fefe:5/64`	

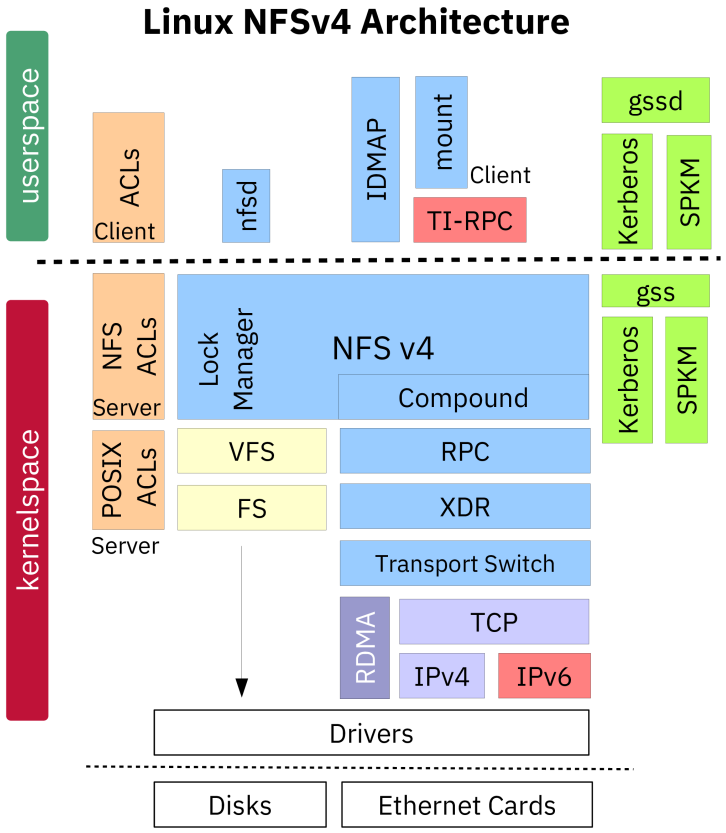
Avant de traiter les questions des sections suivantes, il faut rechercher dans le document *Infrastructure* les éléments nécessaires au raccordement des machines virtuelles ou physiques.

3. Protocole NFSv4

Cette section reprend les éléments spécifiques au protocole NFS introduits lors de la présentation *Systèmes de fichiers réseau*.

Développé à l'origine par Sun Microsystems en 1984, le *Network File System* (NFS) est un protocole de système de fichiers distribué. Il permet de partager des fichiers de manière transparente entre les systèmes hôtes d'un réseau. Cette technologie s'est imposée comme un standard de fait dans le monde Unix pour l'accès aux fichiers distants dans les infrastructures d'entreprise.

Plusieurs versions du protocole de système de fichiers réseau NFS sont disponibles. Chacune correspond à une « époque » ou à un cas d'usage. Le schéma ci-dessous illustre la répartition des fonctionnalités de la version 4 entre les espaces noyau et utilisateur.



Architecture NFS - vue complète

3.1. Architecture NFSv4.2

NFSv4.2 repose sur une architecture unifiée qui combine les espaces utilisateur et noyau, avec une communication strictement sur le port TCP/2049. Cette architecture inclut :

idmapd

Service d'association des identifiants utilisateurs (UID/GID) entre systèmes hétérogènes. Les UID/GID sont représentés sous forme de chaînes, ce qui garantit une indépendance totale des correspondances entre systèmes via le service idmapd.

nfsd

Le démon serveur est traité dans le noyau. Il maintient pour chaque client une arborescence virtuelle cohérente. Il n'est donc plus nécessaire d'exporter les sous-arborescences séparément : tout point accessible permet la navigation sur tous les niveaux inférieurs.

mount

Outil de montage du système de fichiers exporté.

ACLs POSIX/NFS

Prise en charge native des contrôles d’accès.

RPCSEC_GSS et Kerberos

Authentification cryptographique forte et protection contre l'usurpation.

Opérations composées (*compound operations*)

Le regroupement des opérations atomiques en une seule requête permet de réduire la latence et le trafic.

Copie côté serveur (*Server-Side Copy*)

NFSv4.2 permet de copier des fichiers directement sur le serveur, sans transiter par le client, ce qui élimine le trafic réseau redondant pour les opérations en volume (*bulk-data*).

Sparse Files et gestion avancée de l'espace

Le support natif des fichiers dispersés et les opérations ALLOCATE/DEALLOCATE optimisent la réservation et la dissémination de l'espace de stockage afin d'assurer une gestion efficace des ressources.

Parallélisation pNFS

Cette extension du protocole permet d'accéder de manière distribuée et simultanée aux données sur plusieurs serveurs ou chemins, assurant ainsi une évolutivité et des performances linéaires en fonction du nombre de serveurs et d'interfaces réseau.

3.2. Sécurité NFSv4.2

À l'instar de son concurrent, le système de fichiers SMB, la sécurisation des échanges de données en transit a beaucoup progressé. Voici une liste des fonctions les plus importantes :

Labeled NFS

Intégration native avec SELinux pour une gestion des labels de sécurité par fichier.

Chiffrement TLS : NFSv4.2 over TLS

Depuis 2022, il est possible d'utiliser le chiffrement TLS de bout en bout pour l'ensemble des communications NFS, grâce à la norme RPC-over-TLS (RFC 9289).

TLS 1.3 obligatoire

TLS 1.3 renforce la sécurité des communications en réduisant le nombre d'échanges nécessaires pour établir une session chiffrée. Il supprime les algorithmes obsolètes, limite les risques d'attaques et garantit une confidentialité accrue grâce au chiffrement des négociations.

Authentification par certificats x509

Certificats serveurs et clients pour l'identification mutuelle et le chiffrement de bout en bout.

3.3. Adoption NFSv4.2 et innovations

Le marché des systèmes de fichiers réseau affiche une croissance remarquable, avec un chiffre d'affaires évalué à 5,13 milliards de dollars en 2024, puis à 11,24 milliards de dollars en 2032, soit un taux de croissance annuel composé (TCAC) de 12,8 %. Cette expansion s'explique notamment par l'adoption massive du cloud computing. NFSv4.2 s'est imposé comme une solution de choix dans cet écosystème, avec une augmentation de 28 % des déploiements NFS dans les environnements cloud d'une année sur l'autre en 2023. Les principaux fournisseurs de services cloud, comme Amazon Web Services (AWS) avec EFS, Microsoft Azure avec ses implémentations NFS ou encore Google Cloud Platform avec Filestore, ont massivement adopté NFSv4.2 pour leurs offres de stockage géré.

NFSv4.2 révolutionne les cas d'usage traditionnels grâce à ses capacités parallèles avec pNFS (Parallel NFS), qui sont particulièrement adaptées aux charges de travail d'intelligence artificielle et de calcul haute performance. Le protocole trouve de nouveaux débouchés dans l'orchestration de conteneurs avec Kubernetes, où il s'intègre nativement via les StorageClass pour fournir un stockage dynamique partagé.

4. Configuration commune au client et au serveur NFS

Plusieurs services communs doivent être actifs pour que les accès au système de fichiers réseau NFS soient utilisables. Le mécanisme de gestion des appels de procédures distants appelé RPC ou *Remote Procedure Call* constitue le point de départ dans la mise œuvre de ces services communs.

Le logiciel de gestion des appels de procédures distants a évolué avec les différentes versions du système de fichiers NFS et l'arrivée du protocole réseau IPv6. La configuration étudiée ici doit permettre de fonctionner de la façon la plus transparente possible avec les versions 3 et 4 du système de fichiers NFS.

> Note
>
> Les manipulations présentées ici ne traitent pas le volet authentification et chiffrement des échanges sur le réseau. On considère que les services ***Kerberos***, SPKM-3 et LIPKEY ne sont pas actifs sur les systèmes étudiés.

4.1. Gestion des appels RPC

Q1. Quels sont les deux logiciels disponibles chargés de la gestion des appels RPC ? Qu'est-ce qui les distinguent ?

La présentation *Systèmes de fichiers réseau* introduit les principes de fonctionnement des appels de procédures distants.

Rechercher dans le support *Linux NFS-HOWTO* le service «historique» utilisé par NFS pour le multiplexage des appels de procédures distants.

Le support *Linux NFS-HOWTO* présente le service «historique» utilisé par NFS pour le multiplexage des appels de procédure distants : `portmap`. Ce service est fourni par le paquet du même nom et est limité au protocole réseau IPv4.

Le démon `rpcbind` actuel est aussi fourni par le paquet du même nom. C'est un logiciel de multiplexage des appels de procédure distants qui se veut plus évolutif que le précédent et qui supporte le protocole réseau IPv6.

Q2. Quel est le paquet qui correspond à la gestion des appels de procédure distants ?.

Utiliser les outils de recherche dans les répertoires de noms de paquets et dans leurs descriptions : apt-cache, dpkg, aptitude.

Comme indiqué dans la documentation, on recherche un paquet portant le nom `rpcbind`.

```
apt search rpcbind
En train de trier... Fait
Recherche en texte intégral... Fait
rpcbind/testing 1.2.6-6+b1 amd64
  conversion de numéros de programmes RPC en adresses universelles
```

```
sudo apt install rpcbind
```

Q3. Quel est le numéro de port utilisé par le service ? Quel est le principe de fonctionnement du service pour le traitement des appels de procédures distants ?

Utiliser les commandes qui permettent d'obtenir les informations sur :

- La liste des processus actifs sur le système,
- Les numéros de ports en écoute sur les interfaces réseau,
- Les pages de manuels des applications utilisées.
- La liste des processus actifs sur le système,

```
ps aux | grep rpc[b]ind
root      2963  0.0  0.0  18956   724 ?        Ss   14:01   0:00 /sbin/rpcbind -w
```

- Les numéros de ports en écoute sur les interfaces réseau,

```
sudo lsof -i | grep rpc[b]ind
rpcbind   2096        _rpc    4u  IPv4  18957      0t0  TCP *:sunrpc (LISTEN)
rpcbind   2096        _rpc    5u  IPv4    713      0t0  UDP *:sunrpc
rpcbind   2096        _rpc    6u  IPv6   1752      0t0  TCP *:sunrpc (LISTEN)
rpcbind   2096        _rpc    7u  IPv6  20601      0t0  UDP *:sunrpc
```

On obtient la correspondance entre numéro de port et nom de service en consultant le fichier `/etc/services`.

```
grep sunrpc /etc/services
sunrpc          111/tcp         portmapper      # RPC 4.0 portmapper
sunrpc          111/udp         portmapper
```

Le principe de fonctionnement des appels de procédure distants veut que tous ces appels soient reçus sur un numéro de port unique : `sunrpc/111`. Ces appels, une fois identifiés, sont transmis aux programmes concernés pour être traités.

- Les pages de manuels des applications utilisées.

```
man rpcbind
```

Q4. Quelle est a commande qui permet de lister les services accessibles via un appel RPC ? À quel paquet appartient cette commande ?

Rechercher dans le support *Linux NFS-HOWTO* et dans la liste des fichiers du paquet sélectionné pour la gestion des appels RPC.

La commande présentée dans le support *Linux NFS-HOWTO* est appelée rpcinfo. On vérifie sa présence sur le système étudié de la façon suivante.

```
dpkg -S $(which rpcinfo)
rpcbind: /usr/sbin/rpcinfo
```

C'est l'option `-s` qui permet d'obtenir la présentation la plus synthétique des services accessibles par appel RPC.

```
rpcinfo -s
   program version(s) netid(s)                          service     owner
    100000  2,3,4     local,udp,tcp,udp6,tcp6          portmapper  superuser
```

La copie d'écran ci-dessus montre que le gestionnaire d'appel `portmapper` est le seul service ouvert. On relève l'ordre de priorité des différentes versions du service supportées par le système ainsi que les versions des protocoles de couche transport.

Q5. Donner deux exemples d'exécution de la commande pour lister le(s) service(s) ouvert sur le système local puis sur le système voisin.

Reprendre la commande utilisée dans la question précédente en indiquant l'adresse IPv4 ou IPv6 du système voisin.

L'exemple d'exécution de la commande en local est donné dans la copie d'écran de la question précédente. Pour connaître les services accessibles sur un autre poste, on utilise la même commande suivie de l'adresse IP de cet hôte.

```
rpcinfo -s 192.168.51.194
   program version(s) netid(s)                          service     owner
    100000  2,3,4     local,udp,tcp,udp6,tcp6          portmapper  superuser
```

```
rpcinfo -s fe80::baad:caff:fefe:2
   program version(s) netid(s)                          service     owner
    100000  2,3,4     local,udp,tcp,udp6,tcp6          portmapper  superuser
```

Ces copies d'écran montrent la même liste de paramètres que lors de l'exécution de la commande en local. Les configurations sur les deux hôtes sont donc identiques à ce stade de la configuration.

Q6. Réaliser une capture à l'aide de l'analyseur réseau lors de l'exécution de la commande et relever : le protocole de transport utilisé, les numéros de ports caractéristiques de cette transaction ainsi que le nom de la procédure RPC utilisée.

```
système 1                                        système 2
-------------------------------------------------------
 <commande>       --- requête --->            <processus>

                  <-- réponse ----
```

Voici un exemple de capture en mode console qui donne les éléments demandés.

Note

Pour effectuer des captures de trafic réseau en mode console, on dispose de deux applications : tshark et termshark. Pour limiter les dimensions des copies d'écran, on privilégie l'utilisation de tshark.

Pour utiliser l'une ou l'autre des deux applications en tant qu'utilisateur normal, il est nécessaire d'appartenir au groupe `wireshark`. Pour ajouter le compte `etu` au groupe système, on exécute l'instruction `sudo adduser etu wireshark`. Il ne faut pas oublier de se déconnecter puis se reconnecter pour bénéficier de l'attribution au groupe.

Pour une requête IPv4, on obtient :

```
tshark -i enp0s1
Capturing on 'enp0s1'
192.168.51.195 → 192.168.51.194 TCP 74 53284 → 111 [SYN] Seq=0
192.168.51.194 → 192.168.51.195 TCP 74 111 → 53284 [SYN, ACK] Seq=0 Ack=1
192.168.51.195 → 192.168.51.194 TCP 66 53284 → 111 [ACK] Seq=1 Ack=1
192.168.51.195 → 192.168.51.194 Portmap 110 V3 DUMP Call
192.168.51.194 → 192.168.51.195 TCP 66 111 → 53284 [ACK] Seq=1 Ack=45
192.168.51.194 → 192.168.51.195 Portmap 754 V3 DUMP Reply (Call In 4)
192.168.51.195 → 192.168.51.194 TCP 66 53284 → 111 [ACK] Seq=45 Ack=689
192.168.51.195 → 192.168.51.194 TCP 66 53284 → 111 [FIN, ACK] Seq=45 Ack=689
192.168.51.194 → 192.168.51.195 TCP 66 111 → 53284 [FIN, ACK] Seq=689 Ack=46
192.168.51.195 → 192.168.51.194 TCP 66 53284 → 111 [ACK] Seq=46 Ack=690
```

Pour une requête IPv6 avec l'adresse unique, on obtient :

```
tshark -i enp0s1
2001:678:3fc:1f5:baad:caff:fefe:3 → 2001:678:3fc:1f5:baad:caff:fefe:2 TCP 94 51134 → 111 [SYN] Seq=0
2001:678:3fc:1f5:baad:caff:fefe:2 → 2001:678:3fc:1f5:baad:caff:fefe:3 TCP 94 111 → 51134 [SYN, ACK] Seq=0 Ack=1
2001:678:3fc:1f5:baad:caff:fefe:3 → 2001:678:3fc:1f5:baad:caff:fefe:2 TCP 86 51134 → 111 [ACK] Seq=1 Ack=1
2001:678:3fc:1f5:baad:caff:fefe:3 → 2001:678:3fc:1f5:baad:caff:fefe:2 Portmap 130 V3 DUMP Call
2001:678:3fc:1f5:baad:caff:fefe:2 → 2001:678:3fc:1f5:baad:caff:fefe:3 TCP 86 111 → 51134 [ACK] Seq=1 Ack=45
2001:678:3fc:1f5:baad:caff:fefe:2 → 2001:678:3fc:1f5:baad:caff:fefe:3 Portmap 774 V3 DUMP Reply (Call In 4)
2001:678:3fc:1f5:baad:caff:fefe:3 → 2001:678:3fc:1f5:baad:caff:fefe:2 TCP 86 51134 → 111 [ACK] Seq=45 Ack=689
2001:678:3fc:1f5:baad:caff:fefe:3 → 2001:678:3fc:1f5:baad:caff:fefe:2 TCP 86 51134 → 111 [FIN, ACK] Seq=45 Ack=689
2001:678:3fc:1f5:baad:caff:fefe:2 → 2001:678:3fc:1f5:baad:caff:fefe:3 TCP 86 111 → 51134 [FIN, ACK] Seq=689 Ack=46
2001:678:3fc:1f5:baad:caff:fefe:3 → 2001:678:3fc:1f5:baad:caff:fefe:2 TCP 86 51134 → 111 [ACK] Seq=46 Ack=690
```

- Le protocole de couche transport utilisé est TCP.
- Le numéro de port utilisé correspond bien au service enregistré `sunrpc/111`.
- Le sous-programme distant appelé est : `Portmap V3 DUMP Call`.

Pour une requête IPv6 avec l'adresse de lien local, on obtient :

```
tshark -i enp0s1 -f "! port 22"
Capturing on 'enp0s1'
    1 0.000000000 fe80::baad:caff:fefe:3 → fe80::baad:caff:fefe:2 Portmap 102 V3 DUMP Call
    2 0.000265556 fe80::baad:caff:fefe:2 → fe80::baad:caff:fefe:3 Portmap 746 V3 DUMP Reply (Call In 1)
2 packets captured
```

Ici, le protocole de couche transport utilisé est UDP. Comme UDP est non orienté connexion, on ne relève aucune trace d'ouverture ou de fermeture de connexion.

On remarque que la copie d'écran ci-dessus utilise une syntaxe de capture qui permet de filtrer tous les segments qui font appel au port numéro `22` qui correspond au service SSH.

```
tshark -i enp0s1 -f "! port 22"
```

Pour exploiter toutes les informations du trafic capturé, il est conseillé de stocker les résultats dans un fichier à l'aide de la syntaxe suivante.

```
tshark -i enp0s1 -f "! port 22" -w /var/tmp/rpcbind.pcap
Capturing on 'enp0s1'
3 ^C
```

Dans ce dernier cas, seul le compte des trames capturées apparaît à la console.

On peut alors transférer le fichier de capture via la commande scp pour une exploitation via l'interface graphique de Wireshark ou afficher les détails directement à la console. Dans l'exemple ci-dessous, on affiche toutes les informations relatives à la première trame capturée.

```
tshark -r /var/tmp/rpcbind.pcap -V -Y "frame.number == 1"
```

4.2. Gestion des paquets NFS

Q7. Quel est le paquet commun au client et au serveur ? Identifier le jeu de commandes fournies par ce paquet.

Rechercher dans la liste des paquets disponibles, ceux dont le nom débute par `nfs`.

```
aptitude search ?name"(^nfs)" | grep -v ganesha
v  nfs-client -
p  nfs-common - NFS support files common to client and server
p  nfs-kernel-server - support for NFS kernel server
v  nfs-server -
p  nfs4-acl-tools - Commandline and GUI ACL utilities for the NFSv4 client
p  nfstrace - NFS tracing/monitoring/capturing/analyzing tool
p  nfstrace-doc - NFS tracing/monitoring/capturing/analyzing tool (documentation)
p  nfswatch - Program to monitor NFS traffic for the console
```

Dans la liste ci-dessus, on identifie le paquet `nfs-common` qui correspond bien aux fonctions communes au client et au serveur NFS.

```
sudo apt install nfs-common
```

Une fois le paquet installé, la liste des programmes fournis par ce paquet est extraite de la liste de ses fichiers à l'aide de la commande suivante.

```
dpkg -L nfs-common | grep bin
/sbin
/sbin/mount.nfs
/sbin/osd_login
/sbin/rpc.statd
/sbin/showmount
/sbin/sm-notify
/usr/sbin
/usr/sbin/blkmapd
/usr/sbin/mountstats
/usr/sbin/nfsidmap
/usr/sbin/nfsiostat
/usr/sbin/nfsstat
/usr/sbin/rpc.gssd
/usr/sbin/rpc.idmapd
/usr/sbin/rpc.svcgssd
/usr/sbin/rpcdebug
/usr/sbin/start-statd
/sbin/mount.nfs4
/sbin/umount.nfs
/sbin/umount.nfs4
```

Dans cette liste, on trouve les commandes de montage, de démontage et de suivi d'état du système de fichiers réseau.

5. Configuration du serveur NFS

Le rôle du serveur NFS est de mettre à disposition sur le réseau une partie de son arborescence locale de système de fichiers. On parle d'«exportation».

> **Note**
> Il existe plusieurs implémentations libres de serveur NFS. On se limite ici à l'utilisation du logiciel lié au noyau Linux.

Q8. Quel est le paquet qui contient les outils nécessaires au fonctionnement du serveur NFS ? Installez ce paquet.

Interroger les méta données du gestionnaire de paquets pour identifier le nom du paquet à installer.

La recherche des mots clés `nfs` et `server` donne les résultats suivants.

```
aptitude search '?and(nfs, server)'
p   nfs-kernel-server      - support for NFS kernel server
v   nfs-server
```

Les informations données par la commande `apt show nfs-kernel-server` permettent de confirmer qu'il s'agit bien du paquet à installer.

```
sudo apt -y install nfs-kernel-server
```

Q9. Quel est le fichier de configuration principal de gestion des exportations NFS ?

Rechercher dans le support *Linux NFS-HOWTO*.

Quelles que soient les versions du protocole, c'est toujours le fichier `/etc/exports` qui est utilisé. Ce fichier est présenté dans le support *Linux NFS-HOWTO*. Le fichier livré avec le paquet contient, en commentaires, deux exemples complets de configuration NFSv3 et NFSv4. C'est ce dernier exemple que l'on adapte pour traiter les questions suivantes.

Q10. Créer le répertoire `/home/exports/home`. Quelles sont les instructions d'exportation à ajouter au fichier de configuration pour ce répertoire ?

Rechercher dans les supports *Linux NFS-HOWTO* et *Nfsv4 configuration*. On peut aussi utiliser les pages de manuels fournies avec le paquet du serveur NFS.

En exploitant la documentation *Nfsv4 configuration* et l'exemple donné dans le fichier de configuration, on applique les instructions de configuration suivantes dans le fichier `/etc/exports`.

```
sudo mkdir -p /home/exports/home
```

```
cat << EOF | sudo tee -a /etc/exports
/home/exports           192.168.51.192/27(rw,sync,fsid=0,crossmnt,no_subtree_check)
/home/exports/home      192.168.51.192/27(rw,sync,no_subtree_check)
EOF
```

```
cat << EOF | sudo tee -a /etc/exports
/home/exports           2001:678:3fc:1f5::/64(rw,sync,fsid=0,crossmnt,no_subtree_check)
/home/exports/home      2001:678:3fc:1f5::/64(rw,sync,no_subtree_check)
EOF
```

Bien sûr, les adresses des réseaux IPv4 et/ou IPv6 doivent être adaptées au contexte.

Les options entre parenthèses sont documentées dans les pages de manuels `exports` : `man 5 exports`. Les éléments de la liste suivante sont extraits de cette documentation.

- `rw` : autoriser les requêtes en lecture et en écriture sur le volume NFS. Le comportement par défaut est d'interdire toute requête qui modifierait le système de fichiers.
- `sync` : ne répondre aux requêtes qu'après l'exécution de tous les changements sur le support réel.
- `fsid=0` : avec NFSv4, un système de fichiers particulier est la racine de tous les systèmes de fichiers partagés. Il est défini par fsid=root ou fsid=0, qui veulent tous deux dire exactement la même chose.
- `crossmnt` : cette option permet aux clients de se déplacer du système de fichiers marqué `crossmnt` aux systèmes de fichiers partagés montés dessus. Voir l'option nohide.
- `no_subtree_check` : cette option neutralise la vérification de sous-répertoires, ce qui a des subtiles implications au niveau de la sécurité, mais peut améliorer la fiabilité dans certains cas. Si un sous-répertoire dans un système de fichiers est partagé, mais que le système de fichiers ne l'est pas, alors chaque fois qu'une requête NFS arrive, le serveur doit non seulement vérifier que le fichier accédé est dans le système de fichiers approprié (ce qui est facile), mais aussi qu'il est dans l'arborescence partagée (ce qui est plus compliqué). Cette vérification s'appelle subtree_check.

Q11. Comment rendre la configuration d'exportation NFS effective ? Comment vérifier que les paramètres actifs sont corrects ?

Rechercher dans la liste des outils fournis avec le paquet `nfs-kernel-server` la commande qui permet de connaître l'état courant des exportations NFS.

On identifie la commande exportfs dans la liste des binaires fournis avec le paquet serveur NFS.

```
dpkg -L nfs-kernel-server | grep bin
/sbin
/sbin/nfsdcltrack
/usr/sbin
/usr/sbin/exportfs
/usr/sbin/rpc.mountd
/usr/sbin/rpc.nfsd
```

Après chaque modification d'un fichier de configuration, il ne faut surtout pas oublier de relancer le service correspondant.

```
sudo systemctl restart nfs-kernel-server
```

```
systemctl status nfs-kernel-server
# nfs-server.service - NFS server and services
     Loaded: loaded (/lib/systemd/system/nfs-server.service; enabled; vendor preset: enabled)
     Active: active (exited) since Sun 2021-08-29 15:47:25 CEST; 10s ago
    Process: 7699 ExecStartPre=/usr/sbin/exportfs -r (code=exited, status=0/SUCCESS)
    Process: 7700 ExecStart=/usr/sbin/rpc.nfsd $RPCNFSDARGS (code=exited, status=0/SUCCESS)
   Main PID: 7700 (code=exited, status=0/SUCCESS)
        CPU: 8ms

août 29 15:47:24 server-nfs systemd[1]: Starting NFS server and services...
août 29 15:47:25 server-nfs systemd[1]: Finished NFS server and services.
```

Enfin, on consulte la liste des entrées exportées via NFS.

```
sudo exportfs
/home/exports   192.168.51.192/27
/home/exports   2001:678:3fc:1f5::/64
/home/exports/home
                192.168.51.192/27
/home/exports/home
                2001:678:3fc:1f5::/64
```

Cette dernière liste est identique à celle produite par la commande showmount côté client NFS.

Q12. Qu'est-ce qui distingue l'exportation d'une arborescence entre les versions 3 et 4 du protocole NFS ?

Rechercher dans les différences relatives à la notion de nommage dans les manipulations proposées dans les supports *Linux NFS-HOWTO* et *Nfsv4 configuration*.

Donner la signification du paramètre `fsid=0` dans la documentation relative à la version 4. Proposer une analogie avec le fonctionnement d'un serveur Web.

Au delà des évolutions du protocole, c'est la cohérence du système de nommage qui distingue la version 4 du système de fichiers réseau. Il s'agit de garantir qu'un objet (fichier ou répertoire) soit représenté de la même manière sur un serveur et sur ses clients.

Dans le contexte de ces travaux pratiques les répertoires utilisateurs doivent être référencés à partir d'une racine nommée `/ahome/`.

Du point de vue infrastructure, l'utilisation de cette référence de nommage unique présente un avantage non négligeable. En effet, les répertoires d'exportation tels qu'ils ont été définis dans le fichier `/etc/exports` donné ci-dessus désignent un espace de stockage physique.

La racine `/ahome/` désigne un espace de stockage logique. Ce schéma de nommage logique doit rester constant alors que les volumes de stockages physique peuvent migrer et se déplacer, être étendus, etc.

Les différences entre les manipulations proposées dans les supports *Linux NFS-HOWTO* et *Nfsv4 configuration* traduisent les différences de conception entre les deux générations du protocole NFS. On peut relever deux paramètres importants sur le serveur.

- L'option `fsid=0`, présente dans le fichier `/etc/exports/`, permet de définir une ***racine de montage*** tout comme on le verrait sur un serveur Web. Le paramètre de configuration `DocumentRoot /var/www` du serveur ***apache2*** désigne la racine à partir de laquelle les pages Web publiées sont référencées. Cette racine est indépendante de l'arborescence du système de fichier local du serveur.

- L'utilisation d'un montage local avec l'option `bind` de la commande mount permet de mettre en cohérence l'arborescence du serveur et de ses clients. Ainsi, le répertoire `/ahome/` présente les mêmes objets que l'on soit connecté sur le serveur ou sur un client. Le schéma de nommage est donc cohérent.

  Le montage local peut se faire manuellement sur le serveur avec la syntaxe suivante.

```
sudo mkdir /ahome
```

```
sudo mount --bind /home/exports/home /ahome
```

  Une fois la configuration validée, on peut intégrer ce montage local dans la configuration système pour que l'opération soit effectuée à chaque initialisation. Il faut alors éditer le fichier de configuration dédié aux montages des volumes locaux du système : `/etc/fstab`.

  Voici comment ajouter l'instruction de montage au fichier `/etc/fstab` du serveur NFS.

```
echo "/home/exports/home     /ahome  none    defaults,bind    0    0" | \
sudo tee -a /etc/fstab
```

```
grep -v ^# /etc/fstab
UUID=8362b3e6-d426-4f1b-93eb-e1efc22f60f4 /       ext4    errors=remount-ro 0    1
UUID=f3e18b95-7430-4fea-ace5-7dd4cea6398a none    swap    sw    0       0
/home/exports/home     /ahome  none    defaults,bind     0       0
```

Q13. Comment créer un compte utilisateur local baptisé `etu-nfs` avec un répertoire utilisateur situé sous la racine `/ahome` ?

Après consultation des pages de manuels de la commande adduser, on dispose des options de création de compte respectant le critère énoncé. L'option `--home` permet de désigner le répertoire utilisateur dans l'arborescence système.

```
sudo adduser --home /ahome/etu-nfs etu-nfs
```

```
id etu-nfs
uid=1001(etu-nfs) gid=1001(etu-nfs) groupes=1001(etu-nfs)
```

Les identifiants numériques `uid/gid` jouent un rôle important dans la suite des manipulations. Voir Section 7, « Gestion des droits sur le système de fichiers NFS ».

Q14. Créer un fichier texte ayant pour propriétaire l'utilisateur `etu-nfs` côté serveur et visualiser son contenu côté client.

Réaliser une capture et relever les numéros de ports caractéristiques de des transactions de montage. Est-il possible de retrouver le contenu du fichier texte dans les données de capture ?

Pour réaliser cette capture, il faut synchroniser les opérations entre les systèmes client et serveur. On commence par le lancement du l'analyseur réseau puis on visualise le contenu du fichier.

Côté serveur NFS, on créé le fichier texte puis on lance la capture réseau.

```
etu@server-nfs:~$ su - etu-nfs
Mot de passe :
etu-nfs@server-nfs:~$ echo "This file is mine" > textfile
etu-nfs@server-nfs:~$ exit
déconnexion
```

```
etu@server-nfs:~$ tshark -i enp0s1 -f "! port 22"
Capturing on 'enp0s1'
2001:678:3fc:1f5:baad:caff:fefe:2 → 2001:678:3fc:1f5:baad:caff:fefe:3 NFS 254 V4 Call GETATTR FH: 0x455db001
2001:678:3fc:1f5:baad:caff:fefe:3 → 2001:678:3fc:1f5:baad:caff:fefe:2 NFS 330 V4 Reply (Call In 3) GETATTR
2001:678:3fc:1f5:baad:caff:fefe:2 → 2001:678:3fc:1f5:baad:caff:fefe:3 TCP 86 883 → 2049 [ACK] Seq=169 Ack=245
2001:678:3fc:1f5:baad:caff:fefe:2 → 2001:678:3fc:1f5:baad:caff:fefe:3 NFS 262 V4 Call ACCESS FH: 0x455db001, [Check: RD LU
2001:678:3fc:1f5:baad:caff:fefe:3 → 2001:678:3fc:1f5:baad:caff:fefe:2 NFS 258 V4 Reply (Call In 6) ACCESS, [Allowed: RD LU
2001:678:3fc:1f5:baad:caff:fefe:2 → 2001:678:3fc:1f5:baad:caff:fefe:3 TCP 86 883 → 2049 [ACK] Seq=345 Ack=417
2001:678:3fc:1f5:baad:caff:fefe:2 → 2001:678:3fc:1f5:baad:caff:fefe:3 NFS 254 V4 Call GETATTR FH: 0x455db001
2001:678:3fc:1f5:baad:caff:fefe:3 → 2001:678:3fc:1f5:baad:caff:fefe:2 NFS 330 V4 Reply (Call In 9) GETATTR
2001:678:3fc:1f5:baad:caff:fefe:2 → 2001:678:3fc:1f5:baad:caff:fefe:3 TCP 86 883 → 2049 [ACK] Seq=513 Ack=661
2001:678:3fc:1f5:baad:caff:fefe:2 → 2001:678:3fc:1f5:baad:caff:fefe:3 NFS 278 V4 Call READDIR FH: 0x455db001
2001:678:3fc:1f5:baad:caff:fefe:3 → 2001:678:3fc:1f5:baad:caff:fefe:2 NFS 1174 V4 Reply (Call In 12) READDIR
2001:678:3fc:1f5:baad:caff:fefe:2 → 2001:678:3fc:1f5:baad:caff:fefe:3 TCP 86 883 → 2049 [ACK] Seq=705 Ack=1749
2001:678:3fc:1f5:baad:caff:fefe:2 → 2001:678:3fc:1f5:baad:caff:fefe:3 NFS 254 V4 Call GETATTR FH: 0x455db001
2001:678:3fc:1f5:baad:caff:fefe:3 → 2001:678:3fc:1f5:baad:caff:fefe:2 NFS 330 V4 Reply (Call In 15) GETATTR
2001:678:3fc:1f5:baad:caff:fefe:2 → 2001:678:3fc:1f5:baad:caff:fefe:3 TCP 86 883 → 2049 [ACK] Seq=873 Ack=1993
2001:678:3fc:1f5:baad:caff:fefe:2 → 2001:678:3fc:1f5:baad:caff:fefe:3 NFS 322 V4 Call OPEN DH: 0x6cceef4e/
2001:678:3fc:1f5:baad:caff:fefe:3 → 2001:678:3fc:1f5:baad:caff:fefe:2 NFS 442 V4 Reply (Call In 18) OPEN StateID: 0x5daa
2001:678:3fc:1f5:baad:caff:fefe:2 → 2001:678:3fc:1f5:baad:caff:fefe:3 TCP 86 883 → 2049 [ACK] Seq=1109 Ack=2349
2001:678:3fc:1f5:baad:caff:fefe:2 → 2001:678:3fc:1f5:baad:caff:fefe:3 NFS 270 V4 Call READ StateID: 0x7dca Offset: 0 Len: 1
2001:678:3fc:1f5:baad:caff:fefe:3 → 2001:678:3fc:1f5:baad:caff:fefe:2 NFS 214 V4 Reply (Call In 21) READ
2001:678:3fc:1f5:baad:caff:fefe:2 → 2001:678:3fc:1f5:baad:caff:fefe:3 TCP 86 883 → 2049 [ACK] Seq=1293 Ack=2477
2001:678:3fc:1f5:baad:caff:fefe:2 → 2001:678:3fc:1f5:baad:caff:fefe:3 NFS 262 V4 Call CLOSE StateID: 0x5daa
2001:678:3fc:1f5:baad:caff:fefe:3 → 2001:678:3fc:1f5:baad:caff:fefe:2 NFS 202 V4 Reply (Call In 24) CLOSE
2001:678:3fc:1f5:baad:caff:fefe:2 → 2001:678:3fc:1f5:baad:caff:fefe:3 TCP 86 883 → 2049 [ACK] Seq=1469 Ack=2593
```

Comme dans les opérations de capture réseau précédentes, il est préférable de stocker les résultats dans un fichier pour les exploiter ultérieurement avec une interface interactive qui permet d'isoler chaque champ de protocole.

Ici, on relève l'utilisation du protocole TCP en couche transport avec le port enregistré `2049/nfs`. Une analyse détaillée de l'appel de procédure `READ` montre que le contenu du fichier texte est bien visible.

6. Configuration du client NFS

Le rôle du client est d'intégrer un accès au système de fichiers d'un hôte distant dans son arborescence locale. On parle de «montage NFS». Dans un premier temps, on teste les opérations de montage manuel. Bien sûr, ces tests ne peuvent aboutir que si une arborescence à été exportée par un serveur.

Ensuite, on teste les opérations de montage automatisées ou ***automontage***. Si le serveur NFS n'est pas encore disponible au moment des tests de montage manuel, il faut préparer les fichiers de configuration du service d'automontage.

6.1. Opérations manuelles de (montage|démontage) NFS

Q15. Quelle est la commande qui permet de tester la disponibilité du service de montage NFS sur un hôte distant ?

Reprendre l'utilisation de la commande qui donne les listes des procédures distantes disponibles. Elle a été identifiée dans la section précédente.

Relativement aux résultats de la section précédente, la liste des services accessibles via RPC sur le serveur NFS s'est étoffée et le service de montage NFS apparaît clairement.

Voici un exemple de résultat utilisant l'adresse IP du serveur NFS.

```
rpcinfo -s fe80::baad:caff:fefe:3
   program version(s) netid(s)                         service     owner
    100000  2,3,4     local,udp,tcp,udp6,tcp6          portmapper  superuser
    100005  3,2,1     tcp6,udp6,tcp,udp                mountd      superuser
    100003  4,3       udp6,tcp6,udp,tcp                nfs         superuser
    100227  3         udp6,tcp6,udp,tcp                -           superuser
    100021  4,3,1     tcp6,udp6,tcp,udp                nlockmgr    superuser
```

Q16. Quelle est la commande qui permet d'identifier l'arborescence disponible à l'exportation depuis le serveur NFS ?

Rechercher dans la liste des commandes du paquet de service NFS commun au client et au serveur.

Dans la liste des commandes fournies avec le paquet `nfs-common`, on trouve un programme appelé showmount. Après consultation des pages de manuels, on relève l'option `-e` qui permet de consulter l'arborescence exportée par un serveur depuis un client. Voici un exemple d'exécution.

```
sudo showmount -e fe80::baad:caff:fefe:3
Export list for fe80::baad:caff:fefe:3:
/home/exports/home 2001:678:3fc:1f5::/64,192.168.51.192/27
/home/exports      2001:678:3fc:1f5::/64,192.168.51.192/27
```

Les résultats de la copie d'écran ci-dessus supposent que le serveur NFS ait déjà été configurer pour exporter le dossier `home`.

La commande showmount ne produit aucun résultat si le serveur NFS n'est pas configuré.

Q17. Quelle est la commande à utiliser pour les opérations de montage manuel ? À quel paquet appartient cette commande ? Cette commande est-elle exclusivement liée au protocole NFS ?

Après avoir consulté le support *Linux NFS-HOWTO*, interroger la base de données des paquets, rechercher dans le contenus des paquets et consulter les pages de manuels.

La documentation indique que c'est la commande mount qui nous intéresse. On effectue ensuite les recherches avec le gestionnaire de paquets.

```
apt search ^mount$
En train de trier... Fait
Recherche en texte intégral... Fait
mount/testing,now 2.37.2-1 amd64  [installé]
  tools for mounting and manipulating filesystems
```

```
dpkg -L mount | grep bin
/bin
/bin/mount
/bin/umount
/sbin
/sbin/losetup
/sbin/swapoff
/sbin/swapon
```

La commande appartient au paquet du même nom. La consultation des pages de manuels `$ man mount` montre que cette commande n'est pas réservée au seul protocole NFS mais à l'ensemble des opérations de montage pour tous les systèmes de fichiers utilisables.

Q18. Créer le répertoire `/ahome` destiné à «recevoir» le contenu répertoires utilisateurs exportés depuis le serveur NFS. Quelle est la syntaxe de la commande permettant de ***monter*** le répertoire exporté par le serveur NFS sur ce nouveau répertoire ?

Rechercher dans le support *Linux NFS-HOWTO*.

Exemple avec l'adresse IPv6 du serveur NFS.

```
sudo mkdir /ahome
```

```
sudo mount [2001:678:3fc:1f5:baad:caff:fefe:3]:/home /ahome
```

```
mount | grep nfs
[2001:678:3fc:1f5:baad:caff:fefe:3]:/home on /ahome type nfs4 \
 (rw,relatime,vers=4.2,rsize=131072,wsize=131072,namlen=255,hard,proto=tcp6,
 timeo=600,retrans=2,sec=sys,clientaddr=2001:678:3fc:1f5:baad:caff:fefe:2,
 local_lock=none,addr=2001:678:3fc:1f5:baad:caff:fefe:3)
```

Exemple avec l'adresse IPv4 du serveur NFS.

```
sudo mkdir /ahome
```

```
sudo mount 192.168.51.195:/home /ahome
```

```
mount | grep nfs
192.168.51.195:/home on /ahome type nfs4 \
 (rw,relatime,vers=4.2,rsize=131072,wsize=131072,namlen=255,hard,proto=tcp,
 timeo=600,retrans=2,sec=sys,clientaddr=192.168.51.194,
 local_lock=none,addr=192.168.51.195)
```

Q19. Réaliser une capture lors de l'exécution de la commande ls -lAh /ahome et relever les numéros de ports caractéristiques de ces transactions. Est-il possible de retrouver les informations échangées dans les données de capture ?

La marche à suivre est identique à celle de la même question côté serveur NFS.

1. On lance la capture de trafic côté serveur NFS.

```
tshark -i enp0s1 -f "! port 22" -w /var/tmp/ls-nfs.pcap
```

2. On exécute la commande ls -lAh /ahome côté client NFS.

3. Retour côté serveur pour exploiter les résultats.

L'analyse montre que le protocole NFS en version 4 utilise bien le mode `COMPOUND` de traitement par lot des appels de procédure distants RPC. On ne relève dans cette capture que les métadonnées système sur les attributs et les permissions relatives à l'arborescence lue.

Si on reprend la même démarche avec la commande cat d'un fichier texte par exemple, le contenu de ce fichier apparaît en clair dans la capture de trafic.

Q20. Quelles ***seraient*** les opérations à effectuer pour configurer le système et rendre un montage NFS statique permanent ?

Rechercher le fichier de configuration système responsable des montages statiques des partitions.

Il est inutile de modifier les fichiers de configuration du système sachant que l'on change de méthode de montage dans la section suivante.

Il faudrait éditer le fichier `/etc/fstab` pour effectuer un montage statique à chaque initialisation du système. On pourrait par exemple insérer une ligne du type suivant à la fin du fichier.

- Avec le protocole IPv4 :

```
192.168.51.195:/home   /ahome   nfs4    0   0
```

- Avec le protocole IPv6 :

```
[2001:678:3fc:1f5:baad:caff:fefe:3]:/home   /ahome   nfs4    0   0
```

Q21. Quelle est la commande à utiliser pour ***démonter*** le dossier `/ahome` ?

Rechercher cette commande dans la liste des outils forunis avec le paquet `mount`.

C'est la commande umount qu'il faut utiliser pour «détacher» un dispositif de stockage du système de fichiers. Dans le cas de cette section, la syntaxe est la suivante.

```
sudo umount /ahome
```

6.2. Opérations automatisées de (montage|démontage) NFS

Dans cette section, on reprend le processus de montage précédent en utilisant le service d'automontage. L'objectif étant de rendre les opérations d'accès au système de fichiers réseau totalement transparentes pour l'utilisateur, le recours au montage manuel doit être évité le plus possible.

Il existe plusieurs implémentations libres pour le service d'automontage. On se limite ici au logiciel lié au noyau Linux.

Avertissement

Les montages manuels et le service d'automontage ne font pas bon ménage ! Il faut absolument démonter tous les systèmes de fichiers NFS avant d'aborder cette partie.

Q22. Quel est le paquet qui contient les outils nécessaires au fonctionnement de l'automontage ?

Rechercher le mot clé automount dans les descriptions du gestionnaire de paquets.

```
aptitude search "?description(automount)"
p   afuse                           - automounting file system implemented in user-space
p   autodir                         - Automatically creates home and group directories
p   autofs                          - kernel-based automounter for Linux
p   autofs-hesiod                   - Hesiod map support for autofs
p   autofs-ldap                     - LDAP map support for autofs
p   fusiondirectory-plugin-autofs   - autofs plugin for FusionDirectory
p   libnss-cache                    - NSS module for using nsscache-generated fi
p   libunix-configfile-perl         - Perl interface to various Unix configurati
p   nsscache                        - asynchronously synchronise local NSS databases
p   pmount                          - mount removable devices as normal user
i   systemd                         - system and service manager
i   systemd-sysv                    - system and service manager - SysV links
p   udevil                          - Alternative storage media interface
p   udiskie                         - automounter for removable media for Python
p   vfu                             - Versatile text-based file-manager
```

Dans le contexte de ces manipulations, c'est le paquet `autofs` qui nous intéresse.

```
sudo apt install autofs
```

Q23. Comment créer un compte utilisateur local baptisé `etu-nfs` avec un répertoire utilisateur situé sous la racine `/ahome` dont les fichiers et répertoires sont placés sur le serveur NFS ?

Après consultation des pages de manuels de la commande adduser, on dispose des options de création de compte respectant les deux critères énoncés. L'option `--home` permet de désigner le répertoire utilisateur dans l'arborescence système et l'option `--no-create-home` évite la création de ce répertoire sur le système local.

```
sudo adduser --no-create-home --home /ahome/etu-nfs etu-nfs
Attention ! Impossible d'accéder au répertoire personnel que vous avez indiqué (/ahome/etu-nfs) : No such file or directory
Ajout de l'utilisateur « etu-nfs » ...
Ajout du nouveau groupe « etu-nfs » (1001) ...
Ajout du nouvel utilisateur « etu-nfs » (1001) avec le groupe « etu-nfs » ...
Le répertoire personnel « /ahome/etu-nfs » n'a pas été créé.
Nouveau mot de passe :
Retapez le nouveau mot de passe :
passwd: password updated successfully
Changing the user information for etu-nfs
Enter the new value, or press ENTER for the default
        Full Name []: Etudiant NFS
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Cette information est-elle correcte ? [O/n]
```

```
id etu-nfs
uid=1001(etu-nfs) gid=1001(etu-nfs) groupes=1001(etu-nfs)
```

Les identifiants numériques `uid/gid` jouent un rôle important dans la suite des manipulations. Voir Section 7, « Gestion des droits sur le système de fichiers NFS ».

Q24. Quels sont les fichiers de configuration du service d'automontage à éditer ou créer pour que l'utilisateur `etu-nfs` ait accès à ses données personnelles ?

Utiliser les fichiers exemples fournis avec le paquet, les pages de manuels associées et créer un fichier spécifique pour la gestion des comptes utilisateurs.

La liste des fichiers du paquet `autofs` montre qu'il existe une page de manuel consacrée au fichier principal de configuration du service : `/etc/auto.master`. Ces informations permettent de configurer un point de montage au dessous duquel doivent se trouver les répertoires utilisateurs. Ces derniers utilisent un fichier de configuration propre : `/etc/auto.home`.

1. On définit la racine de montage `/ahome` dans le fichier de configuration principal `/etc/auto.master`. Cette racine de montage pointe vers le fichier de configuration dédié au montage automatique des répertoires des utilisateurs.

   Après analyse des commentaires présents dans le fichier `/etc/auto.master`, on créé un fichier spécifique à notre contexte dans le dossier `/etc/auto.master.d/` avec le suffixe `.autofs`.

   ```
   echo "/ahome  /etc/auto.home" | \
   sudo tee -a /etc/auto.master.d/ahome.autofs
   ```

2. On créé le fichier `/etc/auto.home` qui utilise une syntaxe particulière pour que le montage du système de fichiers du serveur soit générique et indépendant du nombre des comptes utilisateurs.

   ```
   echo "*    -fstype=nfs4    [2001:678:3fc:1f5:baad:caff:fefe:3]:/home/&" | \
   sudo tee -a /etc/auto.home
   ```

   - Le premier paramètre est le symbole * qui se substitue au nom d'utilisateur : `etu-nfs` dans notre exemple.
   - Le deuxième paramètre `-fstype=nfs4` correspond à une option de montage qui privilégie la version 4 du protocole NFS. Le jeu des options de montage est le même que pour un montage statique.
   - Le troisième paramètre est l'adresse IPv4 ou IPv6 du serveur. Comme on ne dispose pas d'un service DNS à ce stade de la progression des travaux pratiques, on utilise directement les adresses IP.
   - Le répertoire `/home/` correspond à la configuration de l'exportation NFS sur le serveur. Le répertoire `/home/` est situé sous la racine d'exportation qui est uniquement connue du serveur.
   - Le symbole & indique la répétition du premier paramètre : le nom d'utilisateur.

3. Une fois les fichiers de configuration en place, il ne faut pas oublier de redémarrer le service et de contrôler son bon fonctionnement.

```
sudo systemctl restart autofs
```

```
systemctl status autofs
# autofs.service - Automounts filesystems on demand
     Loaded: loaded (/lib/systemd/system/autofs.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2021-08-29 09:42:16 CEST; 51s ago
       Docs: man:autofs(8)
    Process: 8027 ExecStart=/usr/sbin/automount $OPTIONS --pid-file /var/run/autofs.pid (code=exited, status=0/SUCCESS
   Main PID: 8028 (automount)
      Tasks: 4 (limit: 1131)
     Memory: 1.0M
        CPU: 29ms
     CGroup: /system.slice/autofs.service
             └─8028 /usr/sbin/automount --pid-file /var/run/autofs.pid

août 29 09:42:16 client-nfs systemd[1]: Starting Automounts filesystems on demand...
août 29 09:42:16 client-nfs systemd[1]: Started Automounts filesystems on demand.
```

Q25. Quelles sont les conditions à respecter sur le client et le serveur NFS pour que l'utilisateur `etu-nfs` ait la capacité à écrire dans son répertoire personnel ?

Rechercher les attributs d'un compte utilisateur qui correspondent aux propriétés des objets d'un système de fichiers au sens général.

Les identifiants numériques `uid/gid` doivent nécessairement être identiques sur le client et le serveur NFS. Toute la gestion des droits sur le système de fichiers est conditionnée par ces valeurs.

Q26. Comment prendre l'identité de l'utilisateur `etu-nfs` pour tester la validité du montage ?

Cette validation suppose que l'utilisateur puisse atteindre son répertoire et que l'on visualise l'automontage avec les commandes mount et df.

C'est la commande su qui permet de «changer d'identité» sur le système. On l'utilise donc pour prendre l'identité de l'utilisateur dont le répertoire est situé sur le serveur NFS. Pour que l'opération de montage automatique ait lieu, il suffit de se placer dans ce répertoire.

```
etu@client-nfs:~$ su - etu-nfs
etu-nfs@client-nfs:~$ pwd
/ahome/etu-nfs
etu-nfs@:client-nfs~$ df -HT
Sys. de fichiers                                    Type     Taille Utilisé Dispo Uti% Monté sur
udev                                                devtmpfs   495M       0  495M   0% /dev
tmpfs                                               tmpfs      103M    680k  102M   1% /run
/dev/vda1                                           ext4        72G    2,4G   66G   4% /
tmpfs                                               tmpfs      512M       0  512M   0% /dev/shm
tmpfs                                               tmpfs      5,3M       0  5,3M   0% /run/lock
tmpfs                                               tmpfs      103M       0  103M   0% /run/user/1000
[2001:678:3fc:1f5:baad:caff:fefe:3]:/home/etu-nfs nfs4         72G    2,4G   66G   4% /ahome/etu-nfs
```

```
etu-nfs@client-nfs:~$ mount | grep nfs
[2001:678:3fc:1f5:baad:caff:fefe:3]:/home/etu-nfs on /ahome/etu-nfs type nfs4
 (rw,relatime,vers=4.2,rsize=131072,wsize=131072,namlen=255,hard,proto=tcp6,
 timeo=600,retrans=2,sec=sys,clientaddr=2001:678:3fc:1f5:baad:caff:fefe:2,
 local_lock=none,addr=2001:678:3fc:1f5:baad:caff:fefe:3)
```

Bien sûr, ces manipulations ne sont possibles que si la configuration du serveur est effective.

Q27. Réaliser une capture réseau lors de l'exécution des commandes et relever les numéros de ports caractéristiques de ces transactions. Est-il possible de retrouver les informations échangées dans les données de capture ?

La marche à suivre est identique à celle de la même question côté serveur NFS.

7. Gestion des droits sur le système de fichiers NFS

Le contrôle les droits sur les objets de l'arborescence exportée par le serveur NFS est limité au masque de permissions de ces objets. Il est donc important de faire correspondre les identifiants `uid` et `gid` entre le client et le serveur.

Les manipulations suivantes sont à réaliser en «concertation» entre les administrateurs des postes client et serveur. Le compte utilisateur `etu-nfs` doit avoir été créé sur le serveur et sur le client.

> **Note**
> Ces manipulations se font sans système de gestion centralisé de l'authentification. L'utilisation d'un annuaire LDAP pour fournir une base de comptes utilisateurs fait l'objet d'un support de travaux pratiques qui vient après celui-ci. Ce support se concentre sur le volet système de fichiers réseau.

Q28. Quelles sont les valeurs numériques des identifiants `uid` et `gid` du compte utilisateur `etu-nfs` sur le client et sur le serveur NFS ?

Si les valeurs différent entre le client et le serveur, il faut détruire ces comptes utilisateurs et reprendre les options de la commande adduser pour fournir ces valeurs de façon explicite.

L'extrait du résultat de l'instruction `$ sudo adduser --help` ci-dessous montre les options utiles.

```
adduser [--home DIR] [--shell SHELL] [--no-create-home] [--uid ID]
[--firstuid ID] [--lastuid ID] [--gecos GECOS] [--ingroup GROUP | --gid ID]
[--disabled-password] [--disabled-login] USER
   Ajoute un utilisateur normal
```

Reprendre la question sur la création d'un compte utilisateur local dont le répertoire est situé sur le serveur NFS.

Q29. Sur quel poste peut on créer des fichiers et des répertoires avec des masques de permissions ayant d'autres valeurs `uid` et `gid` que celles de l'utilisateur `etu-nfs` ? Quelles sont les options des commandes chmod et chown à utiliser pour réaliser ces opérations ?

Utiliser les pages de manuels des commandes.

C'est sur le serveur que le super utilisateur a la possibilité de créer n'importe quel objet avec n'importe quel propriétaire dans la mesure où le système de fichiers est local et non réseau.

```
etu@server-nfs:~$ sudo touch /ahome/etu-nfs/ThisOneIsMine
etu@server-nfs:~$ sudo chown etu-nfs.etu-nfs /ahome/etu-nfs/ThisOneIsMine
etu@server-nfs:~$ sudo touch /ahome/etu-nfs/ThisOneIs-NOT-Mine
etu@server-nfs:~$ sudo chown 2000.2000 /ahome/etu-nfs/ThisOneIs-NOT-Mine
etu@server-nfs:~$ sudo ls -lh /ahome/etu-nfs/
total 4,0K
-rw-r--r-- 1 etu-nfs etu-nfs 18 29 août  16:15 textfile
-rw-r--r-- 1 etu-nfs etu-nfs  0 29 août  18:32 ThisOneIsMine
-rw-r--r-- 1    2000    2000  0 29 août  18:33 ThisOneIs-NOT-Mine
```

Côté client, les objets créés sont biens visibles et la vue réseau du système de fichiers NFS passe par une correspondance des propriétaires.

```
etu-nfs@client-nfs:~$ id
uid=1001(etu-nfs) gid=1001(etu-nfs) groupes=1001(etu-nfs)
etu-nfs@client-nfs:~$ ls -lh
total 4,0K
-rw-r--r-- 1 etu-nfs etu-nfs 18 29 août  16:15 textfile
-rw-r--r-- 1 etu-nfs etu-nfs  0 29 août  18:32 ThisOneIsMine
-rw-r--r-- 1    2000    2000  0 29 août  18:33 ThisOneIs-NOT-Mine
```

Côté client NFS, les valeurs des identifiants `uid` et `gid` sont correctement restitués et l'utilisateur n'a que le droit de lecture sur le fichier `ThisOneIs-NOT-Mine`.

Q30. Quel est le service qui assure la conformité des identifiants entre serveur et client NFS ?

Reprendre la liste des service RPC actifs sur les deux systèmes.

Le démon `rpc.idmapd` est fourni avec le paquet `nfs-common`.

8. Documents de référence

Systèmes de fichiers réseau : NFS & CIFS

Systèmes de fichiers réseau : présentation des modes de fonctionnement des systèmes de fichiers réseau NFS & CIFS.

Linux NFS-HOWTO

Linux NFS-HOWTO : documentation historique complète sur la configuration d'un serveur et d'un client NFS jusqu'à la version 3 inclue.

Nfsv4 configuration

Nfsv4 configuration : traduction française extraite des pages du projet CITI de l'université du Michigan.