



PROJET SYSG5  
SYSTÈMES D'EXPLOITATION III

RAPPORT DE PROJET : CRÉATION D'UN SERVEUR NAS AVEC  
RASPBERRY PI MODÈLE 3B

*Directeur* : M. WILLEMSE MICHEL

*Professeur* : Mme. BASTREGHI MONICA

*Élève* : M. PIPERS CHRIS G39864

Année 2018 - 2019

# 1 Executive summary

Ce rapport contient les étapes pour créer un serveur NAS à l'aide d'un Raspberry Pi modèle 3B, un disque dur externe et une machine cliente pour tester le fonctionnement du serveur.

Chaque étape de création est effectuée à l'aide de script afin d'automatiser les actions.

La création d'une carte SD ou d'une clé USB bootable (après avoir activé cette option) se fait avec la distribution Raspbian, plus précisément la version Stretch with desktop de Raspbian.

L'installation du logiciel Samba permet la gestion du serveur NAS ainsi que son accessibilité en le configurant.

Pour stocker les données du serveur un disque dur alimenté par une alimentation externe est connecté au Raspberry Pi. Il faut démonter le système de fichier et ensuite formater le disque dur au format ext3, ext4, ? format supporté par Linux et ses distributions.

Pour tester le serveur, il faut configurer la machine, "le client", avec laquelle y accéder. La configuration permet de reconnaître le Serveur NAS présent sur le Raspberry Pi comme un lecteur réseau. ?

## 2 Tables

### Table des matières

<b>1</b>	<b>Executive summary</b>	<b>1</b>
<b>2</b>	<b>Tables</b>	<b>2</b>
<b>3</b>	<b>Introduction</b>	<b>3</b>
<b>4</b>	<b>Matériel requis</b>	<b>4</b>
<b>5</b>	<b>Mise en place d'un environnement de travail</b>	<b>4</b>
5.1	Présentation . . . . .	4
5.2	Création d'une carte micro SD ou clé USB bootable . . . . .	4
5.3	Script pour créer une clé Raspbian bootable pour Raspberry Pi . . . . .	7
5.4	Activer le boot sur clé USB . . . . .	8
5.5	Script pour activer le boot sur USB . . . . .	9
<b>6</b>	<b>Créer un serveur NAS avec le logiciel Samba</b>	<b>9</b>
6.1	Mise à jour du Raspberry Pi . . . . .	9
6.2	Installation du logiciel Samba . . . . .	10
6.3	Script pour la création du Serveur NAS avec installation et configuration du logiciel Samba . . . . .	11
<b>7</b>	<b>Ajouter un device pour stocker les données</b>	<b>12</b>
7.1	Création du dossier accessible via le Serveur NAS . . . . .	12
7.2	Attribution des droits sur le dossier monté du device . . . . .	12
7.3	Monter le ou les device(s) au démarrage du Raspberry Pi . . . . .	12
7.4	Script pour ajouter le device pour stocker les données . . . . .	14
<b>8</b>	<b>Côté client</b>	<b>15</b>
8.1	Se connecter via l'adresse IP . . . . .	15
8.2	Via le nom du Raspberry Pi . . . . .	15
8.3	Script pour monter un lecteur réseau côté client . . . . .	16
<b>9</b>	<b>Conclusion</b>	<b>17</b>
<b>10</b>	<b>Référence</b>	<b>18</b>
<b>11</b>	<b>Annexes</b>	<b>19</b>

### 3 Introduction

Dans le cadre du cours de "Systèmes d'exploitation III", il est demandé de réaliser un projet défini en accord avec le professeur. Cela dans le cadre d'explorer le vaste domaine en lien avec les systèmes d'exploitation, plus particulièrement celui de GNU linux et ses distributions.

Dans ce projet, il a été question de créer un serveur NAS, Network Attached Storage[0], via un Raspberry Pi modèle 3B, une carte micro SD et un disque dur. Pour ce faire, il a fallu créer une carte micro SD bootable, installer le programme pour la gestion du serveur, monter le disque dur et connecter ce dernier au Raspberry Pi. Ce disque dur est utilisé pour le stockage des données qui sont mises sur le serveur. Enfin, du côté client monté un lecteur réseau afin d'accéder au serveur NAS.

Ces différentes étapes sont réalisées à l'aide de scripts pour que tout soit automatisé.

## 4 Matériel requis

- Un Raspberry Pi modèle 3B
- Une alimentation de 2,5 ampère pour Raspberry Pi
- Une carte micro SD de 8Giga minimum
- Une clé USB de 8Giga minimum
- Un disque dur avec alimentation externe [2]
- Un écran
- Un clavier et souris avec connecteur USB
- Un ordinateur avec une distribution Linux
- Un réseau local
- Une connexion internet

## 5 Mise en place d'un environnement de travail

### 5.1 Présentation

À la base, le Raspberry Pi ne contient pas de bios mais une mémoire que l'on nomme "**One Time Programmable**", **OTP**. Cette dernière est programmée pour booter sur carte micro SD. Sur le modèle 3B, il est possible de booter sur une clé Universal Serial Bus[1], USB possédant une plus grande durée de vie grâce au nombre de cycles d'écriture qui est plus grand que celle des carte micro SD.

Ce mode, une fois activé, est irréversible mais il est toujours possible de booter sur carte micro SD par la suite. Ce dernier étant toujours le boot prioritaire. Pour activer le boot sur clé USB, il faut tout de même booter une première fois sur une carte micro SD pour activer cette option. Pour le boot sur clé USB, toutes les clés ne fonctionnent pas. Plusieurs clés ont été testées et une a fini par fonctionner. Suite à des recherches sur cette problématique, aucune explication n'a été trouvée mais certains tutoriels disponibles sur internet conseillent certaines clés qui fonctionnent à coup sûr.

### 5.2 Création d'une carte micro SD ou clé USB bootable

Pour créer une carte micro SD ou une clé USB bootable, il faut tout d'abord télécharger l'image du système d'exploitation Raspbian[3]. Raspbian est un système d'exploitation basé sur Debian GNU/Linux optimisé pour fonctionner sur Raspberry Pi[4]. Sur le site, il faut prendre la version "**Raspbian Stretch with desktop and recommended software**"[5] qui possède un bureau et est complète. Télécharger la distribution à l'adresse :

[https://downloads.raspberrypi.org/raspbian\\_full\\_latest](https://downloads.raspberrypi.org/raspbian_full_latest)

Avec la commande `wget` :

```
wget -O Raspbian.zip https://downloads.raspberrypi.org/raspbian_full_latest -d /home/no-  
mUser/Documents
```

L'option `-d` permet d'enregistrer le fichier téléchargé dans un dossier spécifique, il s'agit ici, du dossier Documents de l'utilisateur courant. La distribution téléchargée est compressée sous

format ZIP, avant de pouvoir l'utiliser, il faut la décompresser. Cela se fait via la commande **unzip** [20] :

```
unzip /home/nomUser/Documents/Raspbian.zip -d /home/nomUser/Documents
```

L'option **-d** permet d'enregistrer le fichier téléchargé dans un dossier spécifique, il s'agit ici, du dossier Documents de l'utilisateur courant. Pour savoir quel device sélectionné, la commande **lsblk** avec l'option **-p** permet de voir le chemin dans le nom des devices. L'option **-o** avec name et size comme attribut permet de conserver que les colonnes name et size. La commande complète est :

```
lsblk -p -o name,size
```

Après avoir sélectionné le device à utiliser, il faut la préparer afin de la rendre bootable. Pour cela, l'utilisation des commandes **umount** [6] et **dd** [21] sont nécessaires. La première permet de démonter le système de fichier du device et la seconde de copier l'image de la distribution avec le paramètre **bs=4M** [7]. Ce paramètre bs permet de choisir la taille des blocs, ici la sélection de 4M c'est-à-dire 4096bytes qui est la taille généralement recommandée. La commande pour démonter le device est :

```
umount /dev/deviceSelectionné 2> /dev/null
```

L'opérateur **2> /dev/null** [8] permet de rediriger les sorties pour les " jetter " afin qu'elles n'affichent pas toutes les informations sur l'exécution de la commande sur le terminal. La commande pour copier l'image de Raspbian est :

```
dd bs=4M if=/home/nomUser/Documents/nonImage of=/dev/deviceSelectionné status=progress conv=fsync
```

L'option **status=progress** permet de voir les statistiques de transfert et l'option **conv=fsync** [13] s'assure que le fichier soit totalement copié sur le device et que tout soit synchronisé.



### 5.3 Script pour créer une clé Raspbian bootable pour Raspberry Pi

```
#!/bin/bash

#-----
# script pour installer raspbian
#-----

function download()
{
    echo -e "download_func"
    echo Ã©tichargement de la distribution Raspbian
    wget -O Raspbian.zip https://downloads.raspberrypi.org/
        raspbian_full_latest /home/mitch/Documents
}

function unz()
{
    echo -e "unzip_func"
    echo Ã©dcompression du zip de la distribution Raspbian
    unzip /home/mitch/Documents/Raspbian.zip -d /home/mitch/
        Documents
}

#Ã©Rcuprer le nom de l'utilisateur courant
user="$(whoami)"

#Ã©Tlcharger la distribution Raspbian si existe pas
(ls "/home/$user/Documents/Raspbian.zip" >> /dev/null && echo
    fichier ditrbution Raspbian.zip Ã©trouv) || ( download )
read -p "Appuyer sur une touche pour continuer..."
echo ""
echo ""

#Ã©Dcompresser l'image de Raspbian
file=$(ls *raspbian*.img)
(ls "/home/$user/Documents/$file" >> /dev/null && echo fichier
    distribution *Raspbian*.img Ã©trouv) || ( unz )
read -p "Appuyer sur une touche pour continuer..."
echo ""
echo ""

#Selection de la devise par l'utilisateur, verif si existe + verif
    la taille si plus grd que 8GiB7
echo ""
lsblk -p -o name,size
echo ""
echo -e "Select the devise or enter \"display devices\" "
echo ""
```



Après avoir exécuté le script, il suffit de mettre la carte SD (ou la clé USB) dans le Raspberry Pi et il démarrera automatiquement le système d'exploitation.

Par défaut, le nom d'utilisateur du Raspberry Pi est " **pi** " et pour mot de passe " **raspberrypi** ".

## 5.4 Activer le boot sur clé USB

Pour activer le boot sur USB, il faut "reprogrammer" l'OTP en ajoutant au fichier **config.txt** qui se trouve dans le dossier **boot** la ligne " **program\_usb\_boot\_mode=1** " [9] . Une fois cette option activée, elle est définitive mais le bootage sur carte micro SD restera le boot prioritaire et toujours utilisable par la suite.

Il faut démarrer une première fois via une carte micro SD pour pouvoir modifier le fichier et activer ce mode. Pour ajouter la ligne pour activer le boot sur USB il faut utiliser la commande **echo** [24] :

```
sudo echo program_usb_boot_mode=1 | sodo tee -a /boot/config.txt
```

Pour vérifier que cette option est bien activée, il faut utiliser la commande :

```
sudo vcgencmd otp_dump | grep 17
```

La valeur alors retournée doit être : **3020000A**.

## 5.5 Script pour activer le boot sur USB

```
#!/bin/bash

# commande pour activer boot USB
confirm=""
echo Voulez vous activer le boot USB? Attention ceci est
    Ã©irversible
echo valider l'activation?_yes/_no
read_confirm
if [ "$confirm" == "yes" ]
then
    sudo echo program_usb_boot_mode=1 | sudo tee -a /boot/config
        .txt
    echo reboot_necessaire
    read -p "Appuyer sur une touche pour Ã©redmmarer"
    sudo reboot
    echo ""
    echo ""
    echo "Ã©Vrifier que l'option est bien activer Ã©aprs le reboot, la
        Ã©rponse doit Ã©atre 3020000A"
    echo "sudo vcgencmd otp_dump | grep 17"
    read -p "Appuyer sur une touche pour continuer ... "

fi
```

## 6 Créer un serveur NAS avec le logiciel Samba

L'utilisation d'un disque dur alimenté par alimentation externe permet de diminuer la consommation énergétique du Raspberry Pi.

### 6.1 Mise à jour du Raspberry Pi

Avant tout de choses, il faut mettre le Raspberry Pi à jour avec les commandes :

```
sudo apt update
```

```
sudo apt upgrade
```

Création des dossiers accessibles sur le NAS Il faut créer des dossiers qui seront accessibles publiquement et privativement sur le NAS. Afin d'y parvenir, il faut créer des dossiers spécifiques avec la commande **mkdir** [22]. Ces dossiers seront créés avec les commandes :

```
sudo mkdir /home/shares
```

```
sudo mkdir /home/shares/public
```

Administrer les groupes et les droits sur le dossier public précédemment créé.

La commande **chown** [10] sert à administrer les groupes, ici attribué au fichier **public** le groupe "**users**".

```
sudo chown -R root :users /home/shares/public
```

L'option **-R** permet d'attribuer ce groupe à tous les fichiers présents dans ce dossier de manière récursive.

La commande **chmod** [11] sert à administrer les droits, ici attribué les droits au fichier public. Les droits sont **lire(r)**, **écrire(w)** et **exécuter(x)** aux groupes **utilisateurs(u)** et **groupe(g)**. Pour les **autres(o)**, les droits attribués sont **lire(r)** et **exécuter(x)**.

```
sudo chmod -R up=rwx,o=rx /home/shares/public
```

L'option **-R** permet d'attribuer ces droits à tous les fichiers présents dans ce dossier de manière récursive.

## 6.2 Installation du logiciel Samba

Le logiciel Samba permet de gérer la mise en réseau d'un ou de plusieurs disque(s) dur et de procurer un accès à ce ou ces derniers depuis un ordinateur ou autres machines connectées au réseau locale. La commande pour installer le **logiciel Samba** :

```
sudo apt install samba samba-common-bin
```

Après avoir installé le logiciel, il faut attribuer les paramètres pour la partie publique du Serveur NAS. Pour ce faire, il faut modifier le **fichier smb.conf** du dossier **/etc/samba** et ajouter les lignes suivantes à la fin de ce fichier via la commande **echo** [24] :

```
echo "[public]
comment= Public Storage
path = /home/shares/public
valid users = @users
force group = users
create mask = 0660
directory mask = 0771
read only = no " » /etc/samba/smb.conf
```

Pour valider ces modifications, il faut **redémarrer le logiciel Samba**, la commande pour cela est :

```
sudo /etc/init.d/samba restart
```

Ajouter un utilisateur à Samba via la commande **smbpasswd**[12] :

```
sudo smbpasswd -a pi
```

L'option **-a** spécifie que l'utilisateur "**pi**" est ajoutée, ensuite un **mot de passe** est demandé pour cet utilisateur.

### 6.3 Script pour la création du Serveur NAS avec installation et configuration du logiciel Samba

```
#!/bin/bash

# mettre a jour le Raspberry Pi
sudo apt update
sudo apt upgrade

#creation des dossiers accessible sur le NAS
sudo mkdir /home/files
sudo mkdir /home/files/public

# attribution du groupe au fichier public
sudo chown -R root:users /home/files/public

#attribution des droit selon les groupes au fichier public
sudo chmod -R ug=rwx,o=rx /home/files/public

#installation du logiciel Samba
sudo apt install samba samba-common-bin

#configurer pour l'accès a la partie public du nass on ajoute ca au
  fichier /etc/samba/smb.conf
echo "[public]
comment=_Public_Storage
path=_/home/files/public
valid_users=_@users
force_group=_users
create_mask_0771
read_only=_no_" >> /etc/samba/smb.conf

#restart Samba pour modifier la configuration
sudo /etc/init.d/samba restart

#ajout d'un utilisateur a Samba
echo "Attention_Ã©reten_le_mot_de_passe_qui_vous_sera_Ã©demand"
read -p "Appuyer_sur_une_touche_pour_continuer_..."
sudo smbpasswd -a pi
```

## 7 Ajouter un device pour stocker les données

Pour stocker les données du serveur, l'utilisation d'une clé USB ou d'un disque dur est possible. Dans ce projet, le choix fût porté sur un disque dur alimenté par une alimentation externe. Après avoir connecté le disque dur au Raspberry Pi, pour savoir quel est le nom attribué à ce device la commande **lsblk** avec l'option **-p** permet de voir le chemin dans le nom des devices. L'option **-o** avec **name** et **size** comme attribut permet de conserver que les colonnes name et size, la commande complète est :

```
lsblk -p -o name,size
```

Le disque dur doit être démonté et formaté avec un système de fichiers connus de Linux, tel que **ext3**, **ext4**, ? La commande pour le démonter est **umount** [6] :

```
umount /dev/deviceSelectionné
```

La commande pour le formater est **mkfs** [23] :

```
sudo mkfs.ext4 /dev/deviceSelectionné
```

### 7.1 Création du dossier accessible via le Serveur NAS

La création du dossier dans lequel sera monté le device accessible via le Serveur NAS. Après la création de ce dossier, il faut lui attribuer le groupe et les droits qui lui sont propres. La commande **mkdir** [22] sert à créer le dossier dans lequel sera monté le device :

```
sudo mkdir /home/shares/public/disk1
```

### 7.2 Attribution des droits sur le dossier monté du device

La commande **chown** [10] permet d'administrer le groupe, ici attribuer au dossier **disk1** le groupe " **users** ".

```
sudo chown -R root :users /home/shares/public/disk1
```

L'option **-R** permet d'attribuer ce groupe à tous les fichiers présents dans ce dossier de manière récursive. La commande **chmod** [11] sert à administrer les droits, ici attribuer les droits au fichier **disk1**. Les droits sont **lire(r)**, **écrire(w)** et **exécuter(x)** aux groupes **utilisateurs(u)** et **groupe(g)**. Pour les **autres(o)**, les droits attribués sont **lire(r)** et **exécuter(x)**.

```
sudo chmod -R ug=rwx,o=rx /home/shares/public/disk1
```

L'option **-R** permet d'attribuer ces droits à tous les fichiers présents dans ce dossier de manière récursive.

### 7.3 Monter le ou les device(s) au démarrage du Raspberry Pi

Le device précédemment monté, une fois le Raspberry éteint et redémarré ce device ne sera pas automatiquement monté. Pour parer ce problème, il faut ajouter au fichier **fstab** du dossier **/etc** du Raspberry Pi la ligne suivante **"/dev/deviceSelectionné /home/shares/public/disk1 auto noatime,nofail 0 0"** via la commande **echo** [24] :

```
echo "/dev/deviceSelectionné /home/files/public/disk1 auto noatime,nofail 0 0" » /etc/fstab
```



## 7.4 Script pour ajouter le device pour stocker les données

```
#!/bin/bash

#Selection de la devise
lsblk -p -o name,label,type,size
dev=""
valid="not"
confirm=""
until [[ "$valid" == "yes" ]]
do
    echo -e "Select_the_devise_or_enter\"display devices\"_"
    read dev
    if [ "$dev" == "display_devices" ]
    then
        clear
        lsblk -p -o name,label,type,size
    else
        if lsblk -f | grep -wq $dev
        then
            echo -e "select_this_device?_" $dev "_yes_"
            for_yes_or_not_for_not"
            read confirm
            if [ "$confirm" == "yes" ]
            then
                valid="yes"
            fi
        else
            echo -e "this_device_not_exist"
        fi
    fi
done

# demonter et formater la device ( format en ext3 ou ext4 )
umount /dev/$dev
mkfs.ext4 /dev/$dev

#creation des repertoires sur le disque et attribuer les groupes et
droits
sudo mkdir /home/files/public/disk1
sudo chown -R root:users /home/files/public/disk1
sudo chmod -R ug=rwx,or=rx /home/files/public/disk1

#Ã©mont le systeme de fichier pour le device Ã©selectionn
sudo mount /dev/$dev /home/files/public/disk1

#montage automatique au demarrage en ajoutant Ã la fin du fichier /
etc/fstab
echo "/dev/sda1_/home/files/public/disk1_auto_noatime,nofail_0_0" >>
```

## 8 Côté client

Pour accéder au Serveur NAS présent sur le Raspberry Pi du côté client, il faut monter un lecteur réseau. Pour cela, il faut configurer le fichier **fstab** présent dans le dossier **/etc** de l'ordinateur ou de la machine voulant se connecter au serveur NAS. Pour le configurer, il nous faut connaître l'adresse IP de notre Raspberry Pi ou son nom. L'adresse IP étant susceptible de changer si elle n'est pas déclarée fixe, il est recommandé d'utiliser le nom du Raspberry Pi. Pour pouvoir configurer cela, il faut télécharger et installer le paquet **cifs-utils** [14] qui permet de monter un partage réseau. La commande pour installer ce paquet est :

```
sudo apt-get install cifs-utils
```

### 8.1 Se connecter via l'adresse IP

Pour connaître l'adresse IP du Raspberry Pi, il suffit de taper dans le terminal la commande **ip addr show** || et de prendre l'adresse présente dans **wlan0, inet 192.168.??? ???**, ou il faut remplacer les "???" par les chiffres indiqués dans le terminal via la commande **ip addr show**. Il faut configurer le fichier **fstab** [18] en y ajoutant la ligne [15][16] **"//192.168.??? ???/public /media/documents cifs username=pi,password=\*\*\*\*,iocharset=utf8,file\_mode=0777,dir\_mode=0777,noperm 0 0"** L'ajout se fait via la commande **echo** [24] :

```
echo " //192.168.??? ???/public /media/documents cifs user-
name=pi,password=****,iocharset=utf8,file_mode=0777,dir_mode =0777,noperm 0 0 "
» /etc/fstab
```

**192.168.??? ???/public** est le dossier accessible publiquement sur le Raspberry Pi grâce au serveur. Le dossier **/media/documents** est le dossier où est monté le réseau de partage. C'est dans ce fichier que seront accessibles les fichiers et/ou dossiers présents sur le Serveur NAS. Dans ce dossier **/media/documents** peut également accueillir les fichiers et/ou dossiers voulant être stockés par le client. Les options de la commande : **Username=pi** , est l'utilisateur ajouté précédemment dans la création du Serveur NAS. **Password =\*\*\*\***, est le mot de passe demandé après l'ajout de l'utilisateur au Serveur NAS. Les 4\* sont à remplacer par le mot de passe qui a été sélectionné lors de la création de l'utilisateur du serveur NAS. **File\_mode=0777** et **dir\_mode=0777** permettent d'avoir les droits lecture (r), écriture (w) et exécution (x) sur les fichiers et dossiers présents sur ce lecteur réseau. **iocharset=utf8** [17], définit l'encodage utilisé pour le montage du lecteur réseau. **noperm** ||, permet de désactiver les autorisations et d'assurer que les utilisateurs locaux peuvent partager les données sur le lecteur réseau. **0 0** [18], le premier 0 correspond au **<dump>**, pour faire des sauvegardes du système de fichiers, la valeur 0 correspond à ignorer le système de fichiers. Le second 0 correspond au **<pass>**, ce paramètre sert à la commande **fsck** [19] et détermine l'ordre dans lequel les systèmes de fichiers vont être vérifiés.

### 8.2 Via le nom du Raspberry Pi

Si rien n'a été changé, le nom du Raspberry Pi est **" RASPBERRYPI "**. Il suffit dès lors, au lieu de noter l'adresse ip après le double slash, de noter le nom du Raspberry Pi. La ligne



a ajouté à **fstab** est alors : **//RASPBERRYPI/public /media/documents cifs username=pi,password=1234,icharset=utf8,file\_mode=0777,dir\_mode=0777,noperm 0 0**

L'ajout se fait via la commande **echo** [24] :

```
echo      "//RASPBERRYPI/public      /media/documents      cifs      user-
name=pi,password=****,icharset=utf8,file_mode=0777,dir_mode  =0777,noperm  0  0  "
» /etc/fstab
```

### 8.3 Script pour monter un lecteur réseau côté client

```
#!/bin/bash

#-----
# Script pour monter un lecteur reseau Ã©cot client pour server NAS
#-----

#installation du paquet cifs-utils si non Ã©install
sudo apt-get install cifs-utils

#ajout de ligne dans le fichier /etc/fstab pour ajouter le lecteur
Ã©rseau
echo "//RASPBERRYPI/public_/media/documents_cifs_username=pi,
password=1234,icharset=utf8,file_mode=0777,dir_mode=0777,noperm,
_0_0_" >> /etc/fstab

#monter le lecteur Ã©rseau
sudo mount -a
```

## 9 Conclusion

La création d'un serveur NAS avec un Raspberry Pi modèle 3B et un disque dur alimenté par alimentation externe se divise en plusieurs étapes.

Tout d'abord, il faut créer une carte micro SD bootable avec la distribution Raspbian. Le Raspberry Pi modèle 3B permet quant à lui de booter sur une clé USB permettant de garantir une plus grande durée de vie du Système grâce au plus grand nombre de cycles d'écriture de ces dernières comparé aux cartes SD.

Après avoir le système d'exploitation fonctionnel en insérant simplement la carte SD (ou clé USB) avec le Système Raspbian, il faut installer le logiciel Samba qui permet de mettre en place un système de réseau de partage. Après l'installation le logiciel, il faut créer les dossiers de partage qui seront disponible sur le Serveur NAS et apporter certaines configurations telles que la création d'un utilisateur et de son mot de passe associé.

Le logiciel permettant la mise en place d'un Serveur NAS une fois installé, il faut lui ajouter le disque dur où seront stockés les fichiers du Serveur. Pour cela, il faut monter le système de fichiers et le formater au format ext4 ( ou ext3, ? ) qui correspond au format utiliser par Linux. Il faut également créer les dossiers où seront stockés les fichiers, accorder les droits et groupes nécessaires et choisis pour ces dossiers.<sup>4</sup> Pour s'assurer que le disque dur soit monté automatiquement il faut éditer le fichier `/etc/fstab` du Raspberry Pi.

Du côté du client, il suffit d'installer le paquet `cifs-utils` pour gérer les lecteurs réseau et d'éditer le fichier `/etc/fstab` pour monter un lecteur réseau.

## 10 Référence

- [0] [https://fr.wikipedia.org/wiki/Serveur\\_de\\_stockage\\_en\\_r%C3%A9seau](https://fr.wikipedia.org/wiki/Serveur_de_stockage_en_r%C3%A9seau))
- [1] <https://www.raspberrypi.org/documentation/hardware/raspberrypi/bootmodes/msd.md>
- [2] <https://knowledge.parcours-performance.com/monter-disque-dur-externe-partage-rasber>
- [3] <https://www.raspberrypi.org/downloads/raspbian/>
- [4] <https://fr.wikipedia.org/wiki/Raspbian>
- [5] <https://www.raspberrypi.org/forums/viewtopic.php?t=204204>
- [6] <https://linux.101hacks.com/unix/umount/>
- [7] <https://doc.ubuntu-fr.org/dd>
- [8] <https://askubuntu.com/questions/350208/what-does-2-dev-null-mean>
- [9] <https://www.framboise314.fr/boot-simplifie-sur-usb-avec-les-raspberry-pi-1-2-et-3/>
- [10] <http://www.octetmalin.net/linux/tutoriels/chown-changer-proprietaire-groupe-fichier.php>
- [11] <https://doc.ubuntu-fr.org/permissions>
- [12] <https://www.samba.org/samba/docs/current/man-html/smbpasswd.8.html>
- [13] <https://unix.stackexchange.com/questions/369884/using-dd-on-mac-os-conv-fsync-equiv>
- [14] <https://www.it-connect.fr/monter-un-partage-cifs-sous-linux/>
- [15] <https://askubuntu.com/questions/313093/how-do-i-mount-a-cifs-share-via-fstab-and-gi>
- [16] [http://linux.jpvweb.com/linux/mesrecetteslinux/doku.php?id=montage\\_samba](http://linux.jpvweb.com/linux/mesrecetteslinux/doku.php?id=montage_samba)
- [17] <https://askubuntu.com/questions/760058/what-is-the-purpose-of-iocharset-utf8>
- [18] <https://wiki.debian.org/fr/fstab>
- [19] <https://doc.ubuntu-fr.org/fsck>
- [20] <https://www.commentcamarche.net/faq/955-linux-compresser-decompresser-un-fichier-zip>
- [21] <https://doc.ubuntu-fr.org/dd>
- [22] [https://www.tutorialspoint.com/unix\\_commands/mkdir.htm](https://www.tutorialspoint.com/unix_commands/mkdir.htm)
- [23] <https://www.computerhope.com/unix/mkfs.htm>
- [24] <https://www.computerhope.com/unix/uecho.htm>

## 11 Annexes

Vous trouverez dans le dossier compressé :

- un dossier "codes" contenant les différents scripts ;
- un dossier "images" contenant les images utilisées pour le rapport ;
- un readme ;
- un aide mémoire pour l'examen ;
- rapport\_SYSG5\_Pipers\_Chris.pdf : le document pdf généré ;
- rapport\_SYSG5\_Pipers\_Chris.tex : le document latex ;