

Sommaire

Notes

Dans ce CR:

- 192.168.101.190 est l'@ IP de la carte ARMADEUS
- 192.168.101.151 est l'@ IP du PC Host

TP1

- Test de la liaison Ethernet entre le PC et la carte Armadeus
- Pilotage carte avec ethernet via Telnet
- Transfert de fichier via TFTP
- Partage de fichier via NFS
- U-boot
- Démarrage du SE via NFS

TP2

- Gestion des programmes C entre ARMADEUS et PC Host
- Make

Ressources

- Rappels commandes linux
- Mémoires

----- TP1 -----

Test de la liaison Ethernet entre le PC et la carte

Config réseau carte Armadeus

```
# ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:1E:AC:00:14:A9
          inet addr:192.168.101.190  Bcast:192.168.101.255  Mask:255.255.255.0
          inet6 addr: fe80::21e:acff:fe00:14a9/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:117 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:9496 (9.2 KiB)  TX bytes:468 (468.0 B)
          Base address:0xb000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

sit0      Link encap:IPv6-in-IPv4
          NOARP  MTU:1480  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

@IP : 192.168.101.190

Pour attribuer une @IP:

```
ifconfig eth0 192.168.101.xxx netmask 255.255.255.0
```

Pour nous xxx=190

Config réseau PC Host

```
gpea@tpsec:~$ ifconfig -a
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:98:9e:b8
          inet adr:10.0.2.15  Bcast:10.0.2.255  Masque:255.255.255.0
          adr inet6: fe80::a00:27ff:fe98:9eb8/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Packets reçus:4401 erreurs:0 :0 overruns:0 frame:0
          TX packets:1025 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          Octets reçus:5953373 (5.9 MB) Octets transmis:87125 (87.1 KB)

enp0s8    Link encap:Ethernet  HWaddr 08:00:27:a9:d2:a5
          inet adr:192.168.101.151  Bcast:192.168.101.255  Masque:255.255.255.0
          adr inet6: fe80::a00:27ff:fea9:d2a5/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Packets reçus:967 erreurs:0 :0 overruns:0 frame:0
          TX packets:36 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          Octets reçus:86766 (86.7 KB) Octets transmis:5121 (5.1 KB)

lo        Link encap:Boucle locale
          inet adr:127.0.0.1  Masque:255.0.0.0
          adr inet6: ::1/128 Scope:Hôte
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          Packets reçus:1278 erreurs:0 :0 overruns:0 frame:0
          TX packets:1278 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          Octets reçus:110382 (110.3 KB) Octets transmis:110382 (110.3 KB)
```

@IP: 192.168.101.151

Le PC Host a 2 cartes réseaux:

- enp0s3: communication internet (NAT et DHCP) -> @IP: 10.0.2.15. **A ne pas modifier**
- enp0s8: communication avec le SE.

Warning

Il faut que l'@ eth0 du SE et l'@ enp0s3 du PC Host soit sur le même réseau local:
Ici c'est 192.168.101.xxx

On configure l'@ IP du PC Host:

```
sudo ifconfig enp0s3 192.168.101.xxx netmask 255.255.255.0
```

Pour nous xxx = 151

Si on a pas enp0s3 quand on fait `ifconfig -a` :

```
sudo dhclient enp0s3
```

Pilotage carte avec ethernet via Telnet

Objectif:

Piloter la carte via un câble ethernet

Protocole Telnet:

- Non sécurisé
- permet d'émuler un terminal à distance: permet donc d'exécuter des commandes sur une machine distante avec une interface sur le poste local.
- Fonctionne dans un environnement client/serveur: ARMADEUS: serveur et PC Host: client

Mise en place

Sur l'ARMADEUS

Warning

Pas conseillé d'utiliser le compte root pour accéder par telnet à un système

Création d'un compte utilisateur

```
adduser lj
```

On remarque dans `/etc/passwd` :

```
# more /etc/passwd
root:x:0:0:root:/root:/bin/sh
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:100:sync:/bin:/bin/sync
mail:x:8:8:mail:/var/spool/mail:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
operator:x:37:37:Operator:/var:/bin/sh
haldaemon:x:68:68:halld:/:/bin/sh
dbus:x:81:81:dbus:/var/run/dbus:/bin/sh
nobody:x:99:99:nobody:/home:/bin/sh
sshd:x:103:99:Operator:/var:/bin/sh
default:x:1000:1000:Default non-root user:/home/default:/bin/sh
lj:x:1001:1001:Linux User,,,:/home/lj:/bin/sh
```

Un nouvel utilisateur Linux a été ajouté.

et dans `/etc/group` :

```
# more /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:
tty:x:5:
disk:x:6:
wheel:x:10:root
dialout:x:20:
utmp:x:43:
staff:x:50:
haldaemon:x:68:
dbus:x:81:
nobody:x:99:
nogroup:x:99:
users:x:100:
default:x:1000:
lj:x:1001:
```

Lancement du serv Telnet sur l'ARMADEUS

En root on lance le service:

```
telnetd
```

Connexion sur le PC Host

On tape la requête:

```
telnetd 192.168.101.num carte ARMADEUS
```

Et on se connecte avec le compte utilisateur.

Transfert de fichier via TFTP

Le TFTP (Trivial File Transfer Protocol) est un protocole de transfert de fichiers léger et simple, utilisé principalement pour transférer des fichiers entre un serveur (le PC Host) et un client (la carte ARMADEUS) sur un réseau.

Il faut que les paquets xinetd et tftpd soit installés sur le PC Host.

Configuration du serveur

On est sur le PC Host

Le fichier de config : `/etc/xinetd.d/tftpd` :

```
service tftp
{
    socket_type      = dgram
    protocol         = udp
    wait            = yes
    user            = root
    server           = /usr/sbin/in.tftpd
    server_args      = /tftpboot
    disable          = no
}
```

Le répertoire `/tftpboot` contient tous les fichiers que le serveur exportera.

Pour créer le dossier: `sudo mkdir /tftpboot`

Et on modifie les droits: `sudo chmod 775`

Warning

A chaque modification de la configuration il faut toujours redémarrer le service

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

Transfert de fichiers

On est sur la carte ARMADEUS

On tape la requête suivante pour récupérer le fichier "hello" dans `/tftpboot` :

```
tftp 192.168.101.151 -r hello -g
```

- -r signifie remote (distant)
- -g signifie get
- -> On souhaite donc **récupérer** un fichier sur un serveur **distant**.

Note

On fait une copie du fichier, le fichier sur le PC Host n'est pas modifié

Partage de fichier via NFS

Le protocole NFS (Network File System) est un protocole de partage de fichiers développé par Sun Microsystems, conçu pour permettre à des systèmes informatiques de **partager** des fichiers et des répertoires sur un réseau local. NFS permet à des ordinateurs clients d'accéder à des fichiers distants sur un serveur comme s'ils étaient stockés localement.

Installation et configuration

On est sur le PC Host

Si NFS est installé, le fichier `/etc/init.d/nfs-kernel-server` doit être présent.

- On crée un répertoire TP1 dans `/home/gpea`

On crée un fichier de configuration exports dans `/etc` et on écrit:

```
/home/gpea/TP1 192.168.101.190(ro,no_root_squash,sync)
```

- `/home/gpea/TP1` est le répertoire de partage
- `ro` : read only
- `rw` : read write (Le client peut modifier le fichier)

On relance le service (pour chaque modifications):

```
sudo /etc/init.d/nfs-kernel-server restart
```

Utilisation du partage

On est sur la carte ARMADEUS

- On crée un point de montage

```
mkdir -p /mnt/host
```

- On rend le montage effectif avec:

```
mount -t nfs 192.168.101.151:/home/gpea/TP1 /mnt/host
```

Le répertoire `/mnt/host` est une sorte de raccourci vers le répertoire distant `/home/gpea/TP1` situé sur le serveur (PC Host)

Si on crée un fichier `hello_nfs` sur le PC Host dans `/home/gpex/TP1`, on peut visualiser le fichier depuis la carte ARMADEUS dans le répertoire `/mnt/host`.

```
# df
Filesystem            1K-blocks      Used Available Use% Mounted on
ubi0:rootfs           229904        4712    225192   2% /
none                   64            0         64   0% /dev
tmpfs                  62644         44     62600   0% /tmp
192.168.101.151:/home/gpex/TP1
                      7557376    941056    6209408  13% /mnt/host
```

Le fichier /home/gpex/TP1/hello_nfs n'est pas présent physiquement sur la carte. En effet, c'est comme un drive. On a accès au répertoire à distance. Si la connexion est coupée ou que le serveur host supprime le fichier, on n'y aurait plus accès (pas de copies).

On enlève le câble réseau et on a plus accès aux fichiers. Si on remet le câble on récupère l'accès. En conclusion, le partage NFS permet l'accès à un dossier à distance tant qu'on est connecté au réseau. Si on perd la connexion, on perd l'accès que l'on récupère dès le retour de cette dernière.

U-boot

U-boot est le BIOS de la carte Armadeus

Les commandes:

- `printenv` : permet d'afficher les variables d'environnement et leur contenu.
- `printenv nomcommande` : permet d'afficher seulement la variable d'environnement.
- `setenv` : permet de définir des variables d'environnement pour la session en cours.
- `saveenv` : permet de sauvegarder ces variables dans la mémoire flash
- `tftp [loadAdress] [bootfilename]` : permet de télécharger des données depuis un hôte.

```
BIOS> printenv ipaddr
ipaddr=192.168.101.190
BIOS> setenv ipaddr 192.168.101.190
BIOS> ping 192.168.101.151
Using FEC device
host 192.168.101.151 is alive
BIOS> setenv ipaddr 192.168.1.200
BIOS> ping 192.168.101.151
Using FEC device
ping failed; host 192.168.101.151 is not alive
BIOS>
```

Après le redémarrage de la carte, la communication avec le PC Host est impossible parce que l'IP de la carte réseau est la mauvaise (192.168.1.200).

`setenv` définit la variable pour la session présente alors que `saveenv` enregistre les variables d'environnement qui sont situées sur la FLASH.

Démarrage du SE via NFS

Jusqu'à présent, notre se a démarré en allant chercher le noyau et le système de fichier en FLASH. Lors de la phase de développement d'un se, on est amené à modifier assez souvent le noyau et le système de fichier. Pour éviter de flasher la mémoire FLASH à chaque fois (prend du temps et réduit la durée de vie de la flash), on préfère laisser le noyau et le système de fichier sur le PC Host et démarrer le système via le protocole NFS, càd

que le système va chercher à la mise sous tension son noyau et son système de fichiers non plus dans la mémoire FLASH mais sur le PC de développement via NFS.

Procédure

Le système de fichier cible pour la carte Armadeus se trouve dans /targetfs (apf27-linux.bin).

Il faut que l'@IP du PC Host et l'@IP de la carte soit sur le même réseau. (setenv)

Il faut que le dossier de partage soit /targetfs et soit en lecture/écriture:

```
gpea@tpsec:/$ cat /etc/exports
/targetfs 192.168.101.190(rw,no_root_squash, sync)
```

On relance le service.

Maintenant sur la carte dans le U-boot, on doit indiquer au bootloader que le noyau et le système de fichier sont sur le serveur. On doit dire :

- Où se trouve le serveur sur le réseau: son @IP
- Où se trouve le système de fichiers sur le serveur: /targetfs

On tape donc:

```
setenv serverip 192.168.101.151
setenv rootpath /targetfs
run nfsboot
```

Conclusion: Pour "envoyer" un fichier sur la carte Armadeus, il suffit de déplacer le fichier dans /targetfs du PC Host.

----- TP2 -----

Gestion des programmes C entre ARMADEUS et PC Host

Code couleur

Couleur	Sens
Vert	Exécutable
Bleu	Annuaire
Magenta	Lien symbolique
Jaune	FIFO

Compilation d'un programme C

On crée un lien (plus pratique) :

```
ln -s /home/adminprof/Tools/host/usr/bin/arm-linux-gcc arm-linux-gcc
```


et on déplace le lien dans `/usr/local/bin` pour y accéder depuis n'importe quel répertoire :

```
mv arm-linux-gcc /usr/local/bin
```

Donc maintenant pour compiler, on aura pas besoin de taper tout le chemin.

Donc pour compiler:

```
arm-linux-gcc -g -o arm_affiche affiche.c
```

- `-g` : option de débogage
- `-o nom` : nom pour le `.exe`

Attention,

- pour compiler pour processeur arm : `arm-linux-gcc`
- pour compiler sur processeur intel : `gcc`
Le jeu d'instruction n'est pas le même.

Make

----- Ressources -----

Rappels commandes Linux

ls

La commande `ls` est utilisée pour lister les fichiers et répertoires dans un répertoire spécifié. Par défaut, elle affiche le contenu du répertoire courant.

```
ls [options] [répertoire]
```

- **-l**: Affiche les détails sous forme de liste pour chaque fichier ou répertoire, y compris les **permissions, le propriétaire, le groupe, la taille, la date de modification et le nom**.
- **-a**: Affiche également les **fichiers et répertoires cachés**, ceux dont le nom commence par un point (`.`).

On peut aussi combiner: `ls -al`

Sortie ls -l

Bien sûr ! Voici une mini documentation sur les permissions affichées lorsqu'on utilise la commande `ls -l` dans un terminal.

Permissions avec la Commande "ls -l":

Lorsque vous utilisez `ls -l` dans un terminal, la sortie affiche une liste détaillée des fichiers et répertoires dans un répertoire, incluant des informations sur les permissions. Voici comment interpréter ces informations :

Structure de la Sortie "ls -l":

La sortie de `ls -l` a une structure qui ressemble à ceci :

```
-rwxr-xr-x 1 utilisateur groupe 4096 fév 25 12:30 fichier
drwxr-xr-x 2 utilisateur groupe 4096 fév 25 12:30 répertoire
```

1. Type de Fichier (1er caractère) :

- **-** : Fichier ordinaire.
- **d** : Répertoire.
- **l** : Lien symbolique.
- **c** : Fichier de périphérique caractère.
- **b** : Fichier de périphérique bloc.
- **p** : Tube nommé (fifo).
- **s** : Socket.

2. Permissions (9 caractères suivants)

3. Nombre de Liens (2e colonne) :

- Indique le nombre de liens vers le fichier ou le répertoire.

4. Propriétaire et Groupe (3e et 4e colonnes) :

- Identifient le propriétaire et le groupe associé au fichier ou au répertoire.

5. Taille (5e colonne) :

- Représente la taille du fichier en octets.

6. Date de Modification (6e et 7e colonnes) :

- Indique la date et l'heure de la dernière modification du fichier ou du répertoire.

7. Nom du Fichier (Dernière colonne) :

- Donne le nom du fichier ou du répertoire.

chmod

La commande "chmod" permet de modifier les permissions d'accès à un fichier ou à un répertoire. Les permissions sont souvent définies pour les propriétaires, les groupes et les autres utilisateurs.

```
chmod [options] mode fichier
```

Avec + - =

- Ajouter la permission d'exécution au propriétaire :

```
chmod u+x fichier
```

- Retirer la permission de lecture au groupe :

```
chmod g-r fichier
```

- Définir les permissions pour tous les utilisateurs (lire et exécuter, mais pas écrire) :

```
chmod a=rx fichier
```

Avec les octales

- Le premier chiffre représente les permissions du **propriétaire**. -> u
- Le deuxième chiffre représente les permissions du **groupe**. -> g
- Le troisième chiffre représente les permissions des **autres utilisateurs**. -> o
- **Read (R) - 4**: Permet la lecture du fichier ou l'accès au répertoire.
- **Write (W) - 2**: Permet la modification du fichier ou la création de fichiers dans un répertoire.
- **Execute (X) - 1**: Permet l'exécution des fichiers ou l'accès aux répertoires.
- Lecture et écriture pour le propriétaire, lecture pour le groupe et les autres :

```
chmod 644 fichier
```

More

Permet de visualiser le contenu d'un fichier sans le modifier. Plus sécurisé que nano ou gedit.

whoami

Permet de savoir qui est l'utilisateur.

su

Changer d'utilisateur

df

Permet de voir les mémoires