Joris	Monnet	
INF3	dlm-b	

Rapport OS TP2 - Kernel:

Introduction:

Le but de ce travail pratique est de compiler un noyau linux et de manipuler son fichier de configuration afin de voir l'incidence que cela a sur le système (Question 9). Ici, nous utilisons une machine virtuelle. Ce TP a été réalisé dans l'ordre des questions du fichier de consignes et de ce fait, le rapport qui suit, suit le même ordre.

1. Installation d'une machine virtuelle :

J'ai fait le choix de reprendre ma première machine utilisée pour le TP1, c'est un Debian 10.5.0 en mode console, nommé Monnet-01.

2. Vérification de la version du noyau :

```
root@Monnet-01:<sup>~</sup>#_upame_-a
Linux_Monnet-01<mark>4.19.0-</mark>12-amd64 #1 SMP Debian 4.19.152-1 (2020-10-18) x86_64 GNU/Linux
```

On voit ici que le noyau actuel de ma machine est le 4.19.0.

3. Téléchargement du noyau le plus récent :

Le noyau le plus récent est le 5.10.2 du 22/12/20 :

```
mainline: 5.10 2020-12-13 [tarball] [pgp] [patch] stable: 5.10.2 2020-12-21 [tarball] [pgp] [patch] stable: 5.9.16 [EOL] 2020-12-21 [tarball] [pgp] [patch]
```

Je m'appuie sur ce fichier du site de kernel pour télécharger et vérifier la signature du kernel : https://www.kernel.org/signature.html

J'installe d'abord curl:

```
root@Monnet-01:~#<mark>apt install curl</mark>
Lecture des listes de paquets... Fait
```

Je télécharge ensuite le kernel sur la machine virtuelle :

Puis la signature :

Suite à une erreur de ma part la machine Monnet-01 a été corrompue en perdant son disque, j'ai donc recommencer sur la machine Monnet-02, et ai fait des snapshots de la machine virtuelle cette fois.

Je décompresse le kernel :

```
root@Monnet–02:~# unxz linux–5.10.2.tar.xz
unxz : linux–5.10 2.tar : Erreur d'écriture : Aucun espace disponible sur le périphérique
```

À la suite d'une discussion avec les collègues, je me rends compte que mon disque est mal monté et manque de place, je corrige et cette fois ça fonctionne, on a bien le .tar :

```
root@Monnet-02:~# ls -l
total 995360
-rw-r--r-- 1 root root 1019238400 déc. 22 22:35 linux-5.10.2.tar
-rw-r--r-- 1 root root 989 déc. 22 22:41 linux-5.10.2.tar.sign
```

Pour vérifier la signature, je télécharge gpg via apt install gnupg2.

Ensuite je vérifie la signature comme indiqué dans le document du site kernel.org :

```
root@Monnet-02:~# gpg2 --verify linux-5.10.2.tar.sign
gpg: les données signées sont supposées être dans « linux-5.10.2.tar »
gpg: Signature faite le lun. 21 déc. 2020 13:32:28 CET
gpg: avec la clef RSA 647F28654894E3BD457199BE38DBBDC86092693E
gpg: Impossible de vérifier la signature : No public key
```

Comme spécifié dans le document je me retrouve confronté à une erreur classique de manque de clé publique, je fais donc :

```
~# gpg2 -–locate–keys torvalds@kernel.org gregkh@kernel.org
gpg: /root/.gnupg/trustub.gpg : base de confiance créée
gpg: clef 38DBBDC86092693E : clef publique « Greg Kroah–Hartman <gregkh@kernel.org> » importée
            Quantité totale traitée : 1
gpg:
gpg:
                             importées :
gpg: clef 79BE3E4300411886 : clef publique « Linus Torvalds <torvalds@kernel.org> » importée
            Quantité totale traitée : 1
gpg:
gpg:
                            importées : 1
     647F28654894E3BD457199BE38DBBDC86092693E
      [ inconnue] Greg Kroah–Hartman <gregkh@kernel.org>
rsa4096 2011–09–23 [E]
       rsa2048 2011-09-20 [SC]
     ABAF11C65A2970B13OABE3C479BE3E430O411886
      [ inconnue] Linus Torvalds <torvalds@kernel.org>
rsa2048 2011–09–20 [E]
```

Afin de récupérer les clés publiques de Linus Torvalds et Greg Kroan-Hartman.

Ensuite j'indique que cette dernière est bonne :

Ensuite je revérifie la signature du fichier, en utilisant tofu pour prendre en compte que la clé est bonne comme spécifié ci-dessus :

```
root@Monnet-02:~# gpg2 --trust-model tofu --verify linux-5.10.2.tar.sign gpg: les données signées sont supposées être dans « linux-5.10.2.tar » gpg: Signature faite le lun. 21 déc. 2020 13:32:28 CET gpg: avec la clef RSA 647F28654894E3BD457199BE38DBBDC86092693E gpg: Bonne signature de « Greg Kroah–Hartman (gregkh@kernel.org) » [totale] gpg: gregkh@kernel.org: Verified 1 signature in the past 4 minutes. Encrypted 0 messages.
```

C'est bon, la signature est vérifiée, c'est Greg Kroah-Hartman qui en est la source. On peut donc maintenant extraire.

4. Extraction:

J'utilise la commande tar -xvf. Après cela, je vérifie que le dossier est bien extrait :

```
root@Monnet-02:~# ls -l
total 995364
drwxrwxr-x 24 root root 4096 déc. 21 13:30 linux-5.10.2
-rw-r--r-- 1 root root 1019238400 déc. 22 22:35 linux-5.10.2.tar
-rw-r--r-- 1 root root 989 déc. 22 22:41 linux-5.10.2.tar.sign
```

On voit bien qu'un nouveau dossier a été créer. Je me place dedans et affiche la liste de son contenu, le dossier a bien été extrait :

```
root@Monnet-02:~# cd linux-5.10.2/
root@Monnet-02:~/linux-5.10.2# ls -l
total 840
```

5. Préparation d'un fichier de configuration :

```
Je prépare un fichier.config: root@Monnet-02:~/linux-5.10.2# touch .config
```

Finalement cette solution ne convient pas, en effet partir d'un fichier de config vide complexifie la partie suivante utilisant make menuconfig, après divers essais, j'utilise donc finalement cette commande: \$ cp -v /boot/config-\$(uname -r) .config afin de récupérer le fichier de config de mon ancien noyau.

6. Modification du fichier de configuration :

```
J'installe make avec: root@Monnet-02:~/linux-5.10.2/net# apt install make
```

Ensuite j'installe gcc, pkg-config et libncurses-dev, bison, flex, utiles pour la commande make. Puis, je fais un make menuconfig pour commencer la configuration du kernel avec menuconfig, j'arrive sur cette page :

```
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----). Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes
features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in
[] excluded <M> module <> module capable
              General setup
           [*] 64−bit kernel
                Processor type and features --->
               Power management and ACPI options --->
               Bus options (PCI etc.) --->
               Binary Emulations --->
                irmware Drivers --->
           [*] Virtualization --->
               General architecture-dependent options --->
           [*] Enable loadable module support --->
           [*] Enable the block layer -
               IO Schedulers
                xecutable file formats --->
               Memory Management options --->
           [*] Networking support --->
               Device Drivers --->
File systems --->
           Security options --->
-*- Oryptographic API --->
Library routines --->
               Kernel hacking --->
                   <Select>
                                 < Exit >
                                               < Help >
                                                             < Save >
                                                                           < Load >
```

Pour la modification de uname je vais dans general settings -> Local version – append to kernel release :

	Local version – append to kernel release Please enter a string value. Use the <tab> key to move from the input field to the buttons below it.</tab>	
	joris	
	<mark>⟨ Ok ></mark> ⟨ Help >	

Pour la suppression du protocole ipv6, je vais dans Networking, j'exclus le protocole IPV6 :

```
Arrow keys navigate the menu. 〈Enter〉selects submenus ---〉(or empty submenus ----).
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes
features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in
[] excluded <M> module <> module capable
         [*]
                 IP: multicast policy routing
         [*]
                 IP: PIM-SM version 1 support
                 IP: PIM-SM version 2 support
         [*]
               IP: TCP syncookie support
         -*-
                √irtual (secure) IP: tunneling
         {M}
               IP: Foo (IP protocols) over UDP
               IP: FOU encapsulation of IP tunnels
         [*]
                IP: AH transformation
         <M>:
              IP: ESP transformation
         <M>
               IP: ESP transformation offload
         <M>
                 IP: ESP in TCP encapsulation (RFC 8229) (NEW)
              IP: IPComp transformation
         <M>
         <M>
              INET: socket monitoring interface
                UDP: socket monitoring interface
         \langle M \rangle
                 RAW: socket monitoring interface
         <M>:
                 INET: allow privileged process to administratively close sockets
               TCP: advanced congestion control
         [*]
                CP: MD5 Signature Option support (RFC2385)
         [*]
              The IPv6 protocol
               NetLabel subsystem support
               MPTCP: Multipath TCP (NEW)
             Security Marking
             Timestamping in PHY devices
                <Select>
                            < Exit >
                                         < Help >
                                                     < Save >
                                                                 < Load >
```

Je me suis aidé de ce site :

https://www.linuxtopia.org/online books/linux kernel/kernel configuration/ch09s04.html

Je sauvegarde, puis fait un cat dans le .config, le fichier a bien été généré. On voit que les variables de configurations ont bien été ajoutées.

Après plusieurs essais avec différentes configurations qui ne marchaient pas (ne boot pas au démarrage), j'arrive enfin à une version viable à partir du fichier de config de l'ancien noyau.

7. Compilation et ajout du noyau au GRUB:

Maintenant que notre fichier de config est créé, on peut compiler avec make -j 5 pour utiliser plusieurs de mes cœurs de processeur (sans ça la commande make durait environ 4h30) :

```
error: Cannot generate ORC metadata for CONFIG_UNWINDER_ORC=y, please install libelf–dev, libelf–dev
el or elfutils–libelf–devel
make: *** [Makefile:1227: prepare–objtool] Error 1
```

Suite à cette erreur je télécharge les packages demandés ave apt install. Ensuite je recommence le make. Je m'aide de ce site https://www.cyberciti.biz/tips/compiling-linux-kernel-26.html. Une fois la commande make terminée, l'image est prête :

```
Kernel: arch/x86/boot/bzImage is ready (#1)
```

Ensuite je fais make modules install pour installer tous les modules nécessaires.

Ensuite, j'installe le kernel afin de mettre les fichiers nécessaires dans le répertoire /boot :

Le make install ajoute tout seul le nouveau kernel a GRUB.

8. Redémarrage et vérification de version et d'IP:

Après avoir redémarré je fais un uname -a :

```
root@Monnet-02:~# uname -a
Linux Monnet-02 5.10.2joris #5 SMP Thu Dec 24 04:34:47 CET 2020 x86_64 GNU/Linux
```

On voit le résultat attendu, c'est-à-dire la version suivie de mon prénom et la date de compilation. Puis je vérifie la pile IP (en ayant au préalable installer net-tools) :

```
root@Monnet-02:~# ifconfig -a
enp0s3: flage-4163 UP, PROADCAST, RUNNING, MULTICAST> mtu 1500
inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
ether 08:00:27:8c:f5:15 txqueuelen 1000 (Ethernet)
RX packets 196 bytes 261201 (255.0 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 126 bytes 9178 (8.9 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flage-73/UP, LOOPBACK, RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
loop txqueuelen 1000 (Boucle locale)
RX packets 1 bytes 29 (29.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 1 bytes 29 (29.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Il n'y a pas d'ipv6, c'est tout bon.

9. Modification du fichier de configuration :

Je décide pour commencer d'autoriser les headers de kernel dans /sys/kernel/kheaders.tar.xz :

```
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----).
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes
features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in
[] excluded <M> module <> module capable
            Enable process_vm_readv/writev syscalls
         [*]
            uselib syscall
         -*- Auditing support
             RQ subsystem --->
             imers subsystem –
             reemption Model (Voluntary Kernel Preemption (Desktop)) --->
             PU/Task time and stats accounting
         [*] CPU isolation
             CU Subsystem
          > Kernel .config support
        Enable kernel headers through /sys/kernel/kheaders.tar.xz
              Kernel log buffer size (16 => 64KB, 17 => 128KB)
         (12) CPU kernel log buffer size contribution (13 => 8 KB, 17 => 128KB)
         (13) Temporary per-CPU printk log buffer size (12 => 4KB, 13 => 8KB)
             cheduler features
         [*] Memory placement aware NUMA scheduler
               Automatically enable NUMA aware memory/task placement
             Control Group support –
         [*] Namespaces support
         [*] Checkpoint/restore support
         [*] Automatic process group scheduling
            Enable deprecated sysfs features to support old userspace tools
         -*- Kernel->user space relay support (formerly relayfs)
                <Select>
                            < Exit >
                                                    < Save >
                                        < Help >
                                                                < Load >
```

Je refais, le make, le make modules_install, le make install et le reboot. Ensuite je fais un cd /sys/kernel puis ls -l :

```
root@Monnet-02:/sys/kernel# 1s -1
total O
                                              30 21:14 boot_params
               2 root root
2 root root
drwxr-xr-x
                                     O déc.
                                     o déc.
                                              30 21:14 cgroup
driiixr-xr-x
                                              30 21:11 debug
             31 root root
                                     O déc.
                                              30 21:14 fscaps
30 21:14 iommu_groups
                                 4096 déc.
               1 root root
driiixr-xr-x
                                               30 21:14 irq
x-rx-rxwrb
                 root root
                                     O déc.
                                              30 21:14 kexec_crash_loaded
30 21:14 kexec_crash_size
30 21:14 kexec_loaded
                                 4096 déc.
               1 root root
                                 4096 déc.
4096 déc
               1 root root
               1 root root 3717264 déc. 30 21:14 kheaders.tar.xz
                                               30 21:14 livepaton
30 21:11 mm
                 root root
                                     O déc.
lrwxr−xr−x
                 root root
                                              30 21:14 notes
30 21:14 profiling
r--r--r--
                                  496 déc.
                 root root
                                 4096 déc.
                 root root
                                 4096 déc.
                                               30 21:14 rcu_expedited
                                              30 21:14 rcu_normal
30 21:11 security
riii-r--r--
                                 4096 déc.
                                     O déc.
lrwxr−xr−x
                 root root
                                               30 21:14 slab
drwxr–xr–x 131 root root
                                     O déc.
                                               30 21:14 software_nodes
driiixr-xr-x
                 root root
                                               30 21:11 tracing
lr-xr-xr-x
                 root root
                                 4096 déc.
                                               30 21:14 uevent_seqnum
                 root root
                                              30 21:14 vmcoreinfo
                 root root
                                 4096 déc.
```

On trouve bien le fichier, qui n'est pas vide à l'endroit susmentionné. Afin de montrer la différence, j'affiche dans le même dossier la liste des fichiers avec mon ancienne version de kernel :

```
root@Monnet–02:/sys/kernel# ls –l
total O
drwxr-xr-x
             2 root root
                            O déc.
                                     30 21:20 boot_params
drwxr-xr-x
             2 root root
                            O déc.
                                     30 21:20 cgroup
            29 root root
                            O déc.
                                     30 21:19 debug
druix-----
             1 root root 4096 déc.
                                     30 21:20 fscaps
                                     30 21:20 iommu_groups
             2 root root
                             O déc.
drwxr-xr-x
drwxr−xr−x
            23 root root
                             O déc.
                                     30 21:20 irq
                    root 4096 déc.
                                     30 21:20 kexec_crash_loaded
             1 root
             1
               root root 4096 déc.
                                     30 21:20 kexec_crash_size
rw-r--r--
             1 root root 4096 déc.
                                     30 21:20 kexec_loaded
                             O déc.
                                     30 21:20 livepatch
drwxr-xr-x
             2 root root
drwxr-xr-x
             6 root root
                             O déc.
                                     30 21:19 mm
                          504 déc.
                                     30 21:20 notes
             1 root root
             1 root root 4096 déc.
                                     30 21:20 profiling
                    root 4096 déc.
                                     30 21:20 rcu_expedited
             1 root
             1 root root 4096 déc.
                                     30 21:20 rcu_normal
                            O déc.
             4 root root
                                     30 21:19 security
driiixr-xr-x
                            O déc.
                                     30 21:19 slab
drwxr–xr–x 121 root root
                             O déc.
                                     30 21:19 tracing
             2 root root
             1 root root 4096 déc.
                                     30 21:20 uevent_segnum
             1 root root 4096 déc. 30 21:20 vmcoreinfo
```

On voit que le fichier n'y était pas à la base.

Ensuite, je décide d'enlever l'accès à l'usb :

```
row keys navigate the menu. 〈Enter〉 selects submenus ---〉 (or empty submenus
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in
[] excluded <M> module <> module capable
           [] Board level reset or power off
                  ower supply class support
           {*} Hardware Monitoring support --->
-*- Thermal drivers --->
            [*] Watchdog Timer Support --->
{M} Sonics Silicon Backplane support --->
{M} Broadcom specific AMBA --->
                Multifunction device drivers
           -*- Voltage and Current Regulator Support ---> <M> Remote Controller support --->
              ] HDMI CEC RC integration
                HDMI CEC drivers
            <M>> Multimedia support --->
                 Graphics support
           <M> Sound card support --->
                HID support
           [_] USB support ----
<M> MMC/SD/SDIO card support
           <M> Sony MemoryStick card support --->
                 LED Support
            [*] Accessibility support --->
                InfiniBand support
            <*> EDAC (Error Detection And Correction) reporting --->
                    <Select>
                                    < Exit >
                                                   < Help >
                                                                   < Save >
```

Je refais le make, make modules_install et make install. Ensuite je montre les usbs connectés :

```
root@Monnet_02:~#<mark>lsusb</mark>
root@Monnet_02:~#_
```

Il n'y en a aucun comme prévu. Pour comparer je réalise la même commande dans mon ancien kernel :

```
root@Monnet–02:~# Isusb
Bus 001 Device 002: ID 80ee:0021 VirtualBox USB Tablet
Bus 001 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
root@Monnet–02:~#
```

On voit ici que cette même commande nous montre tous les usbs connectés. Notre changement a donc bien fonctionné.

Finalement, je décide de retirer la possibilité de se connecter aux réseaux TCP/IP :

```
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----).
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes
features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in
[] excluded <M> module <> module capable
         <*> Packet socket
               Packet: sockets monitoring interface
         <*> ⊍nix domain sockets
               UNIX: socket monitoring interface
         <M>:
         [*] XDP sockets
               <u> XDP sockets: mo</u>nitoring interface
         [ ] TCP/IP networking
              ecurity Marking
             Timestamping in PHY devices
         [*] Network packet filtering framework (Netfilter) --->
         <M> Asynchronous Transfer Mode (ATM)
               LAN Emulation (LANE) support
         <M>> 802.1d Ethernet Bridging
               VLAN filtering
         [*]
               MRP protocol
         <M> 802.10/802.1ad VLAN Support
               GVRP (GARP VLAN Registration Protocol) support
               MVRP (Multiple VLAN Registration Protocol) support
         [*]
         <M> DECnet Support
         [ ]
               DECnet: router support
         <M>> ANSI/IEEE 802.2 LLC type 2 Support
         <M> Appletalk protocol support
               Appletalk interfaces support
                <Select>
                            < Exit >
                                        < Help >
                                                    < Save >
                                                                 < Load >
```

Je refais le make, make modules install et make install, puis je reboot.

Une fois le kernel lancé, je fais un ifconfig -a :

On voit qu'il n'y a pas de socket inet disponibles, pour rappel, ce sont ceux-ci qui donnaient les adresses ip dans la partie 8 :

```
root@Monnet-02:~# ifconfig -a
enp0s3: flags=4163/UP_RROADCAST_RUNNING_MULTICAST> mtu 1500
inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
etner U8:00:27:80:15:15 txqueuelen 1000 (Etnernet)
RX packets 196 bytes 261201 (255.0 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 126 bytes 9178 (8.9 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73/UP_LOOPBACK_RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
loop txqueuelen 1000 (Boucle locale)
RX packets 1 bytes 29 (29.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 1 bytes 29 (29.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

L'accès a un réseau TCP/IP est donc bel et bien indisponible.

Conclusion:

Pour conclure, ce TP m'a permis de remettre en pratique mes connaissances dans les commandes linux utilisées au TP1. De plus, il m'est plus simple maintenant de comprendre comment fonctionne le noyau linux et de manière plus générale un système d'exploitation et ses connexions au hardware. La partie 9 m'aura notamment permis de tester les liens entre le kernel et certaines fonctions de bases de toute système d'exploitation comme l'USB ou le TCP/IP.