Copie de la mémoire et du disque via l'UEFI

Par Solal Jacob

Développeur de

Contenu

- La problématique
- BIOS et UEFI
- Bitlocker
- La solution

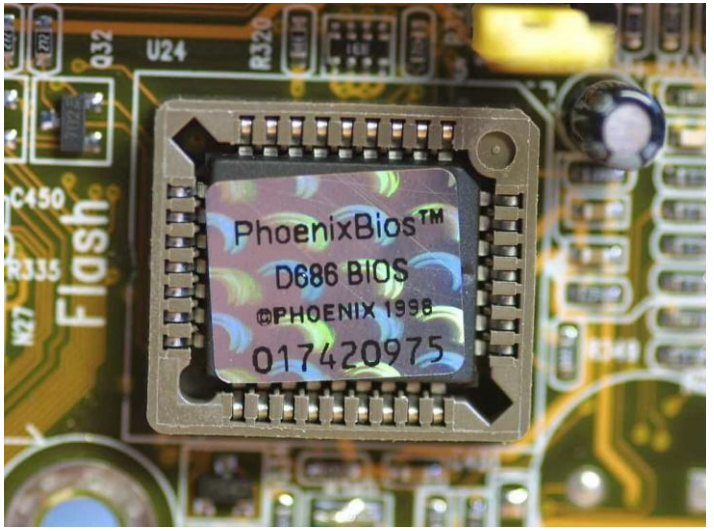
Problématique

Mission d’analyse forensique d'un transformer book :

- Windows 10
- 1 port USB
- SSD soudé
- UEFI 32 bits

Le bios : principe de base

Basic Input Output System, créée en 1975 pour le système CP/M
Première instruction exécutée lorsque l'on appuie sur le bouton "on"



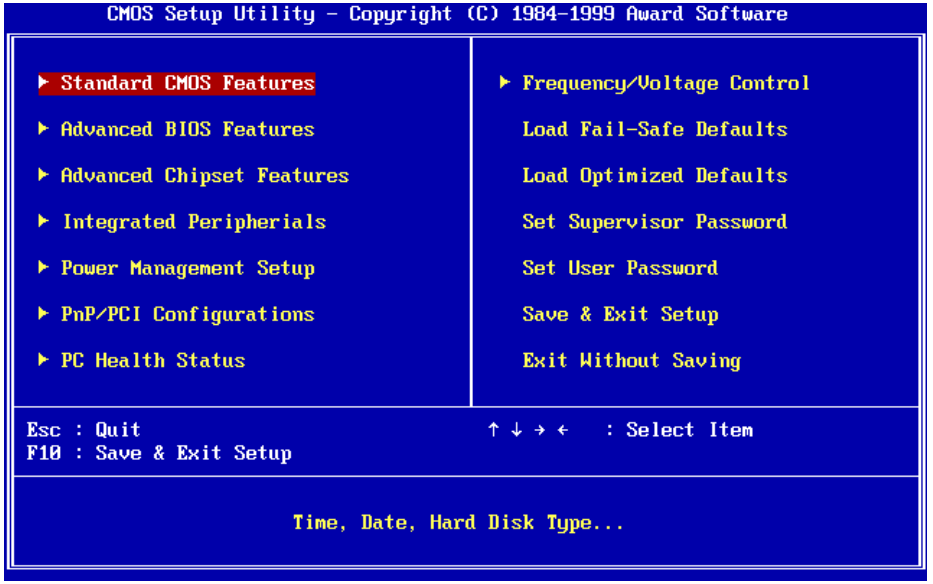
Le code :

- Inscrit dans une mémoire non-volatile (EEPRom en général)
- Dépend du "chip-set" de la carte-mère
- Développé en ASM en général
- En source fermée (Projet coreboot pour un code ouvert)

Le bios : fonctionnement

- Un signal “reset” est envoyé par le “chipset” à la “CPU”
- La CPU démarre et lit ce qui se trouve à l’adresse 0xFFFF0 dans la ROM
- On y trouve une instruction “jump” qui pointe vers le début du code du BIOS
- L’affichage et le clavier sont activés
- Puis les autres périphériques (disque dur, cd-rom, carte réseau, bus USB, …)
- L’interface graphique du BIOS devient accessible

Le bios : interface utilisateur



- Sélectionner un périphérique de démarrage
- Régler l'horloge interne
- Configurer le matériel
- Activer, désactiver les périphériques
- Configurer un mot de passe pour protéger le démarrage ou les paramètres
- Afficher certaines informations système sur les disques, la mémoire, …

Le bios : la séquence de démarrage

Si le périphérique de démarrage est un disque dur :

La “zone d’amorçage” (Master Boot Record) est lue depuis le tout début du disque (premier cylindre, première piste, premier secteur)

- Elle contient le code de démarrage et la table des partitions

Ce code est exécuté et va démarrer :

- Directement le système d’exploitation
- Un “boot loader” (GRUB, LILO, MS)
  qui va permettre de choisir le système a exécuter

La table des partitions

4 Partitions principales contenant :

- Adresse CHS de début et de fin
- Adresse LBA de début
- Nombre de secteurs utilisés par la partition
- Type de partition (FAT, NTFS, EXTFS)
- Flags (bootable ou non)

1 ou des partitions étendues contenant:

- Des partitions secondaires

Supporte une taille maximum de disque de 2 TO

UEFI le remplaçant du BIOS

1998 : Intel crée le standard
Extensible Firmware Interface

2006 : UEFI remplace maintenant EFI

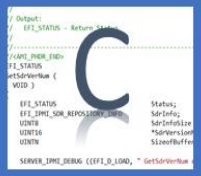
- Une interface logiciel entre le “Firmware” et le système d’exploitation

- Le “U” est apparu quand d’autres industriels se sont joints au standard (AMD, American Megatrends, Apple, Dell, HP, IBM, Insyde, Intel, Lenovo, Microsoft, and Phoenix Technologies)

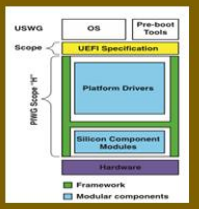
- Peut tourner au-dessus ou a la place du BIOS

- Démarre le système plus rapidement

UEFI : Avantage



Mostly written in C.
High code re-use.

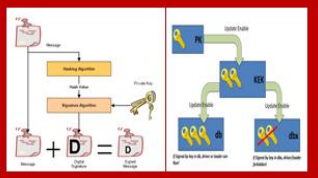


Emphasis on Specifications.
Standards compliance.

Better platform scaling. For e.g. removes shadow ROM limits.

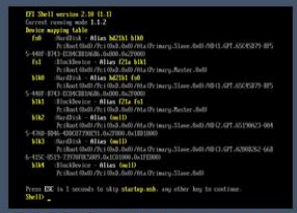
Storage.
GPT removes 2.2 TB MBR restriction.

CPU Architecture independent. Platform design flexibility.



Secure boot solves "trust" related system integration challenges.

Pre-boot Networking.
Ipv4, Ipv6, PXE,
VLAN, iSCSI etc.



UEFI shell improves pre-boot testing & diagnostics experience.

UEFI : Architecture

• SECurity : exécution des processus d'authentification et de contrôle d'intégrité (SecureBoot, mot de passe, token USB)

• Pre Efi Initialization : Initialisation de la carte-mère et du chipset

• Driver eXecution Environment : enregistrement de tous les pilotes

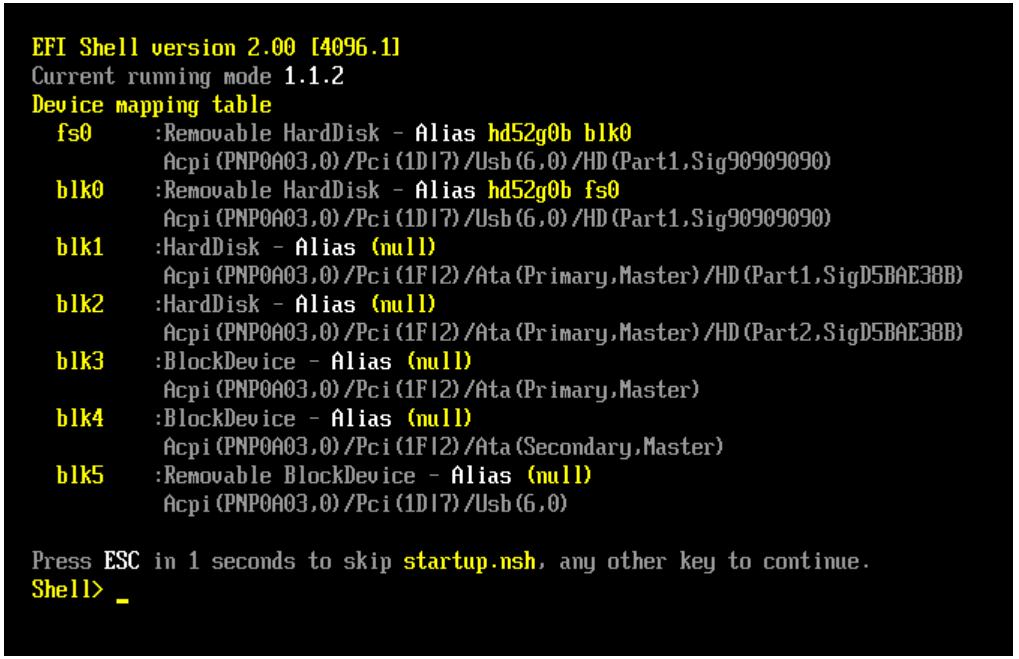
• Boot Device Selection : gestionnaire de démarrage comme grub

• Transient System Load : phase transitoire où le système d'exploitation est chargé

•RunTime : le système d'exploitation a pris la main
Interaction via des variables EFI stockées dans la NVRAM

UEFI : Shell

Développé par Intel
Ressemble à un shell Unix ou DOS
Permet de lancer des applications “UEFI”
Interpréteur python ou ruby

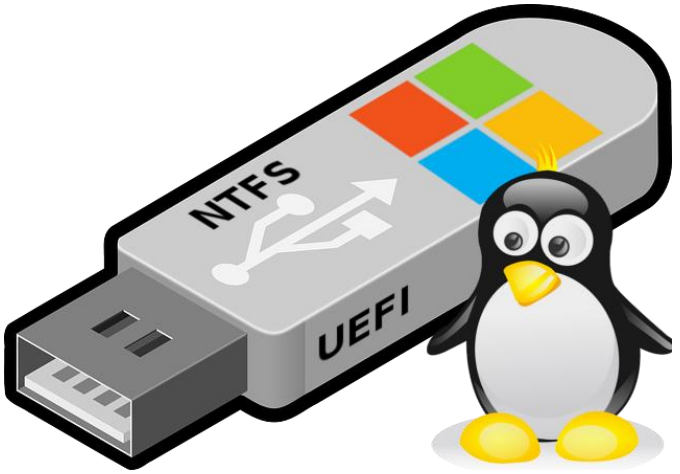


Possède de nombreuses commandes :

- Redirection
- Lister les partitions
- Monter un système de fichiers (fat)
- Info sur la RAM
- Info sur les périphériques
- Editeur hexadécimal (RAM & DISK)

Contient déjà beaucoup de choses utiles pour une analyse forensique
Il peut être chargé via un clef usb ainsi que d’autres applications …

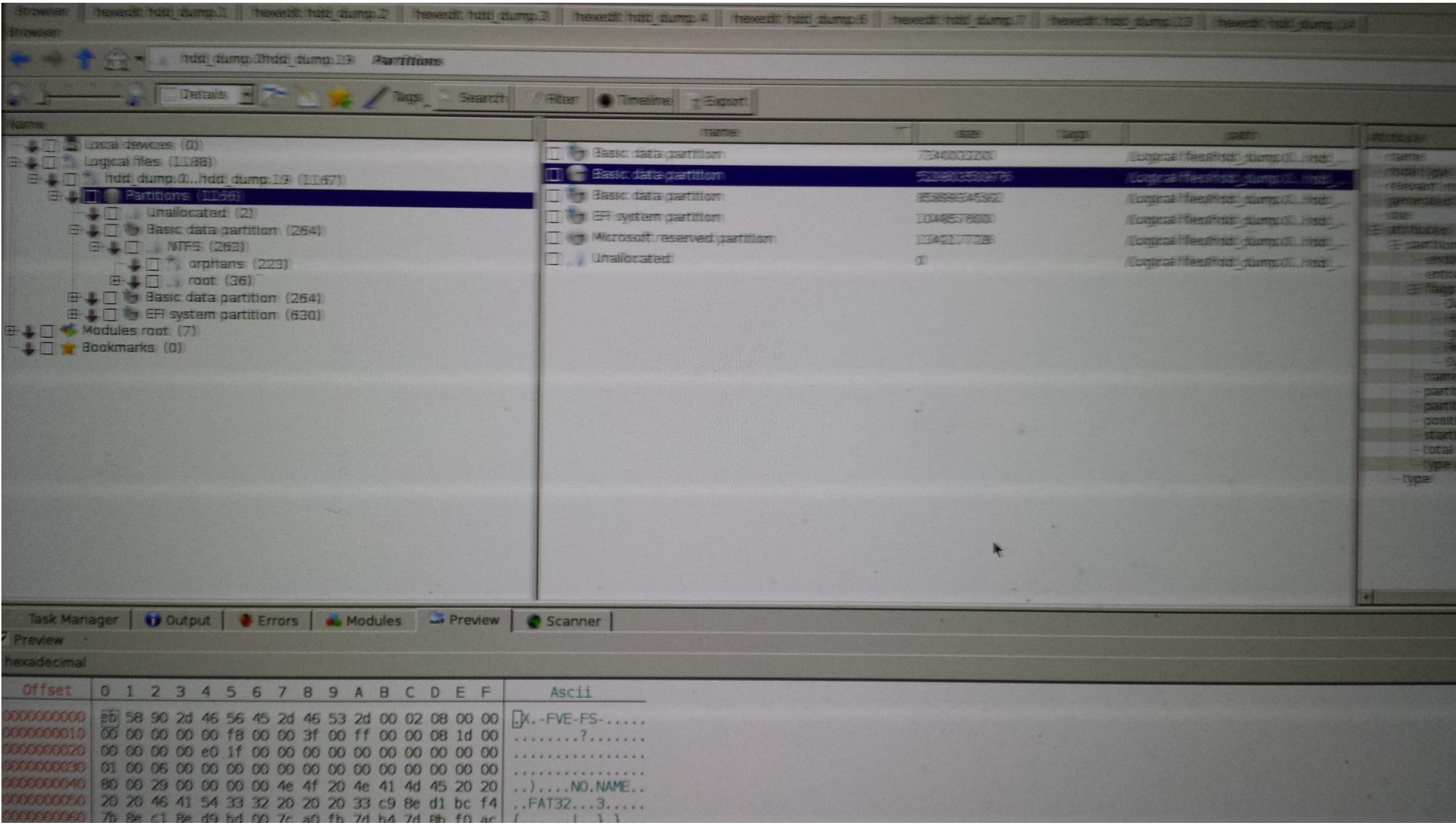
Analyse



Démarrage via Linux (Mint) en USB
Copie du disque :

« Freeze » de Linux (image partielle)

Problème : Chiffré avec Bitlocker



Bitlocker : Fonctionnement

Chiffrement complet du disque

Chiffrement basé sur AES

Stockage des clefs sur TPM, Clef USB, Carte à puce

Deux partitions :

- Partition de démarrage non chiffré
- Partition chiffrée (système de fichiers NTFS)

En-tête NTFS modifiée contenant des infos sur:

- La taille du système de fichiers
- Les métadonnées liées aux différentes clefs
- Clefs du disque chiffrées par la clef utilisateur

Bitlocker : Outils pour déchiffrer

LIBBDE : (Joachim Metz)

Formats supportés :

- Windows Vista
- Windows 7
- Windows 8 (Consumer Preview)
- BitLocker To Go

Méthodes de stockage des clefs supportées:

- Clear key
- Password
- Recovery password
- Start-up key
- FVEK (Full Volume Encryption Key) et TWEAK

Bitlocker : Outils pour déchiffrer

Dislocker : (Romain Coltel)

- Support de Windows 10 AES-XTS
- Permet de récupérer les clefs par défaut
- Information complète sur les métadonnées
- Support de l’écriture
- Code bien structuré et lisible

Bitlocker Attaque : Recherche des clefs en mémoire

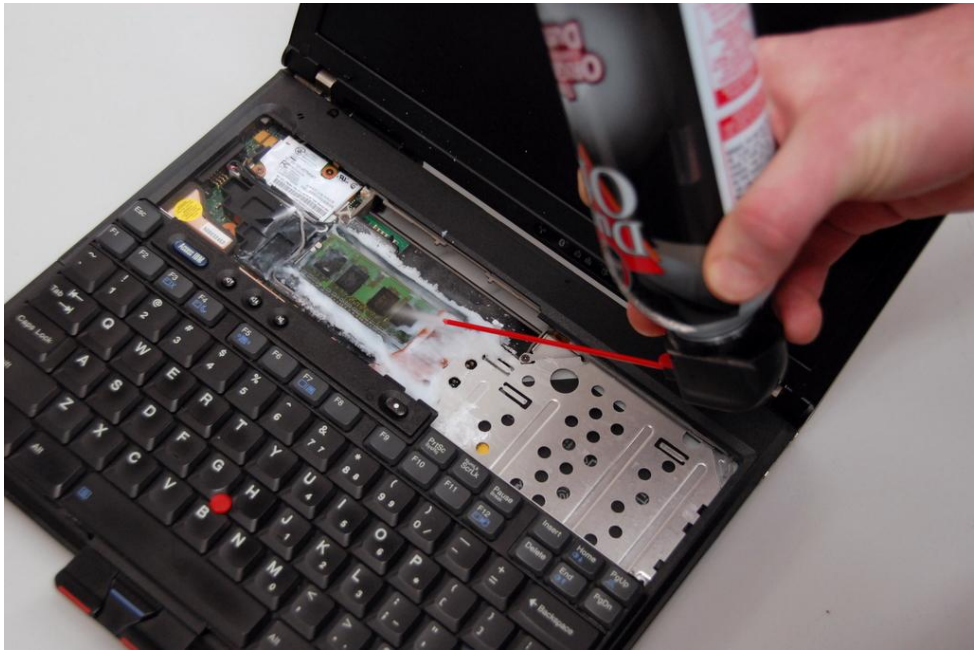
Module volatility bitlocker

Nécessite une copie mémoire compatible avec volatility
Ou un fichier hibernation
Recherche via « pattern » spécifique autour de la clef
(Pool tag : FVEc, Cngb)

AES Key Finder :

Format RAW
Algorithme générique de détection de clefs AES

Copie de la RAM : Attaque « cold boot »



Lest We Remember : Cold Boot Attacks on Encryption Keys
(http://citp.princeton.edu/memory)

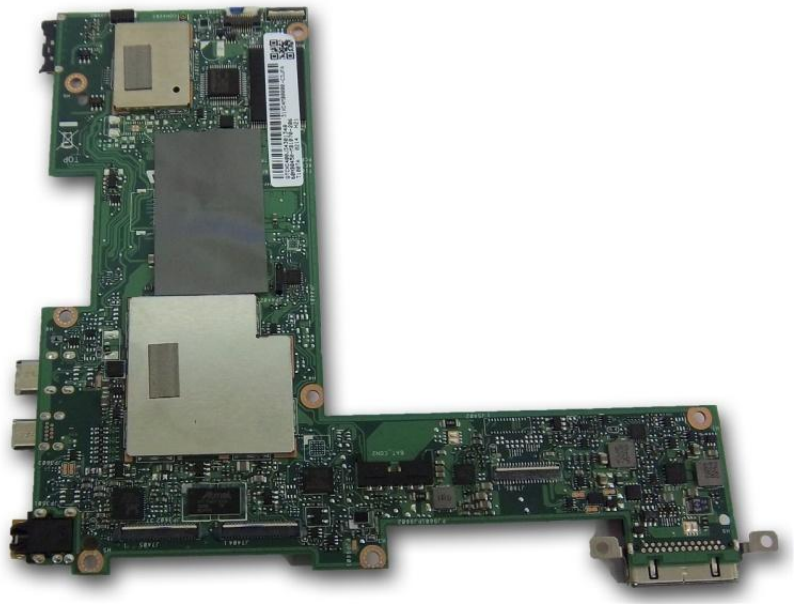
Problème :

- RAM soudée
- Code non maintenu
- Copie sur une clefs usb via BIOS/MBR
- Démarrage EFI copie via PXE

Problématique

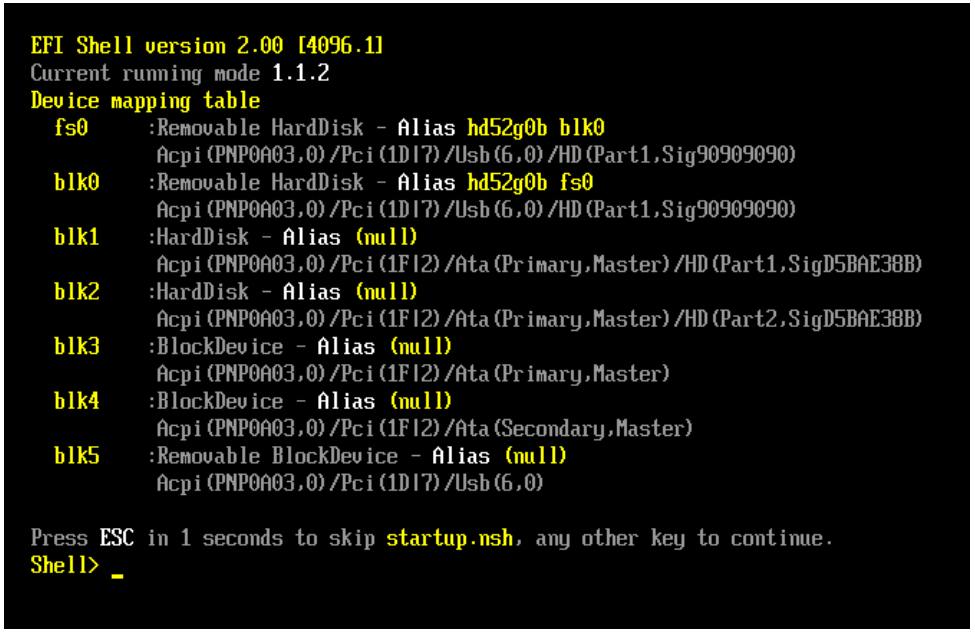
RAM soudée

Pas de copie complète du « SSD »



Solution

Développement d’une application UEFI

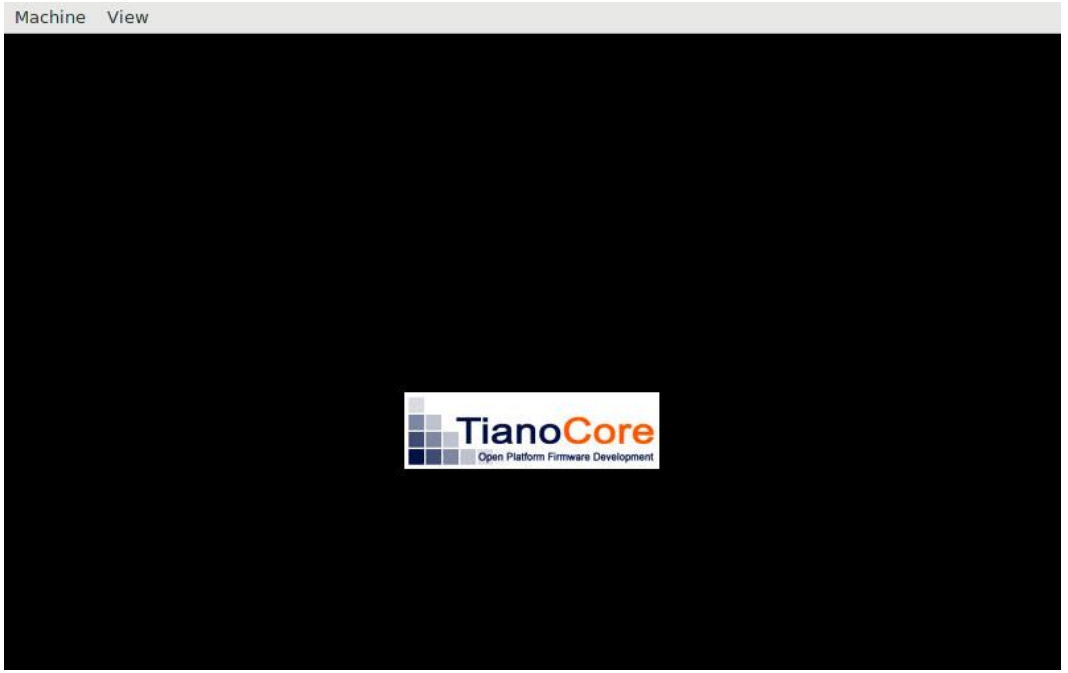


Copie le contenu de la RAM
Copie le contenu du disque
Les fichiers sont placés sur une clef ou un disque USB (FAT)
Découpés en fichiers de 4Go

Solution : outil de développement

Utilisation du SDK Intel Edk2 TianoCore
http://www.tianocore.org/edk2

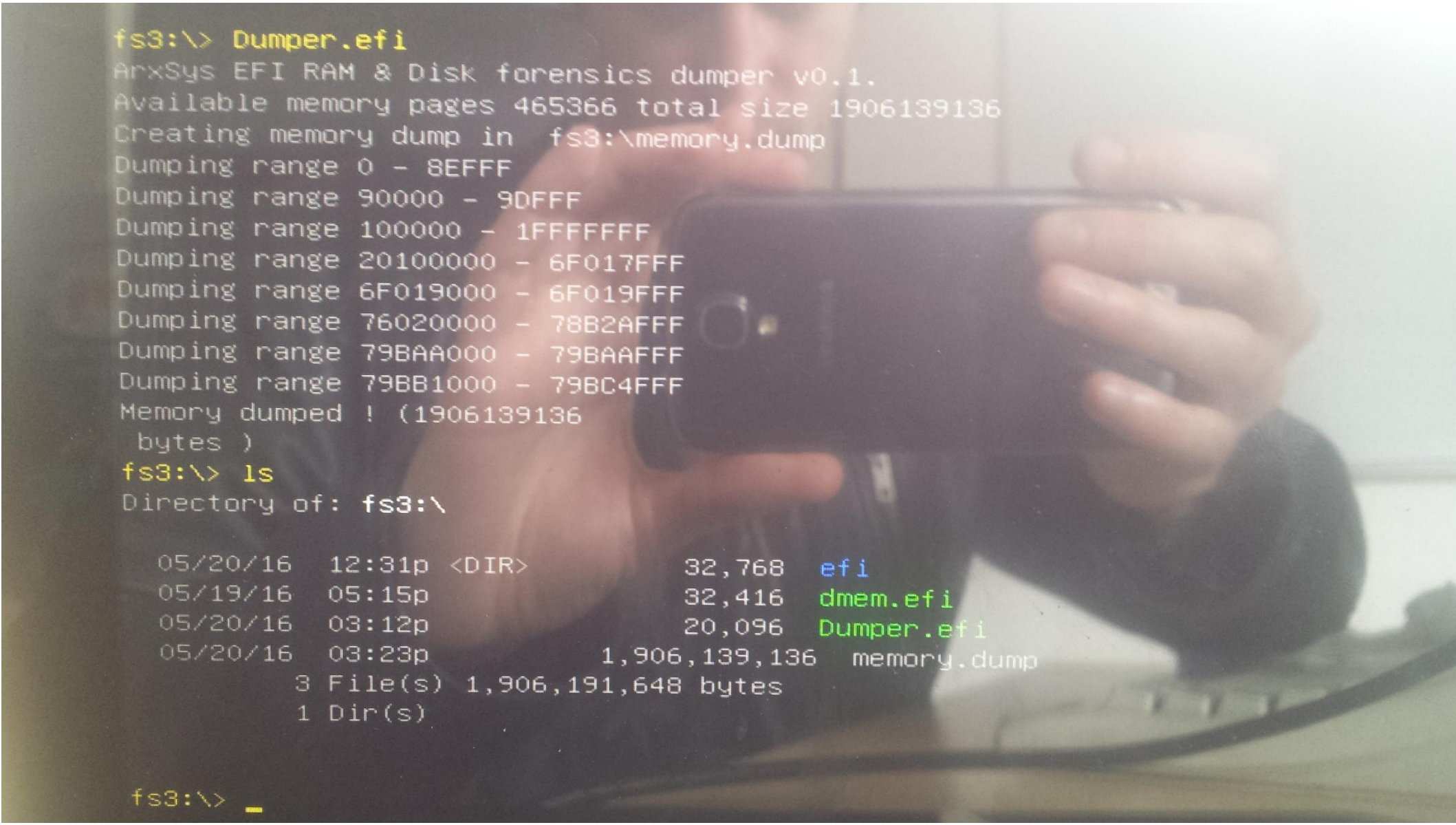
Développement en C
API complète
Génération d'exécutable (PE) 32 & 64 bits & drivers



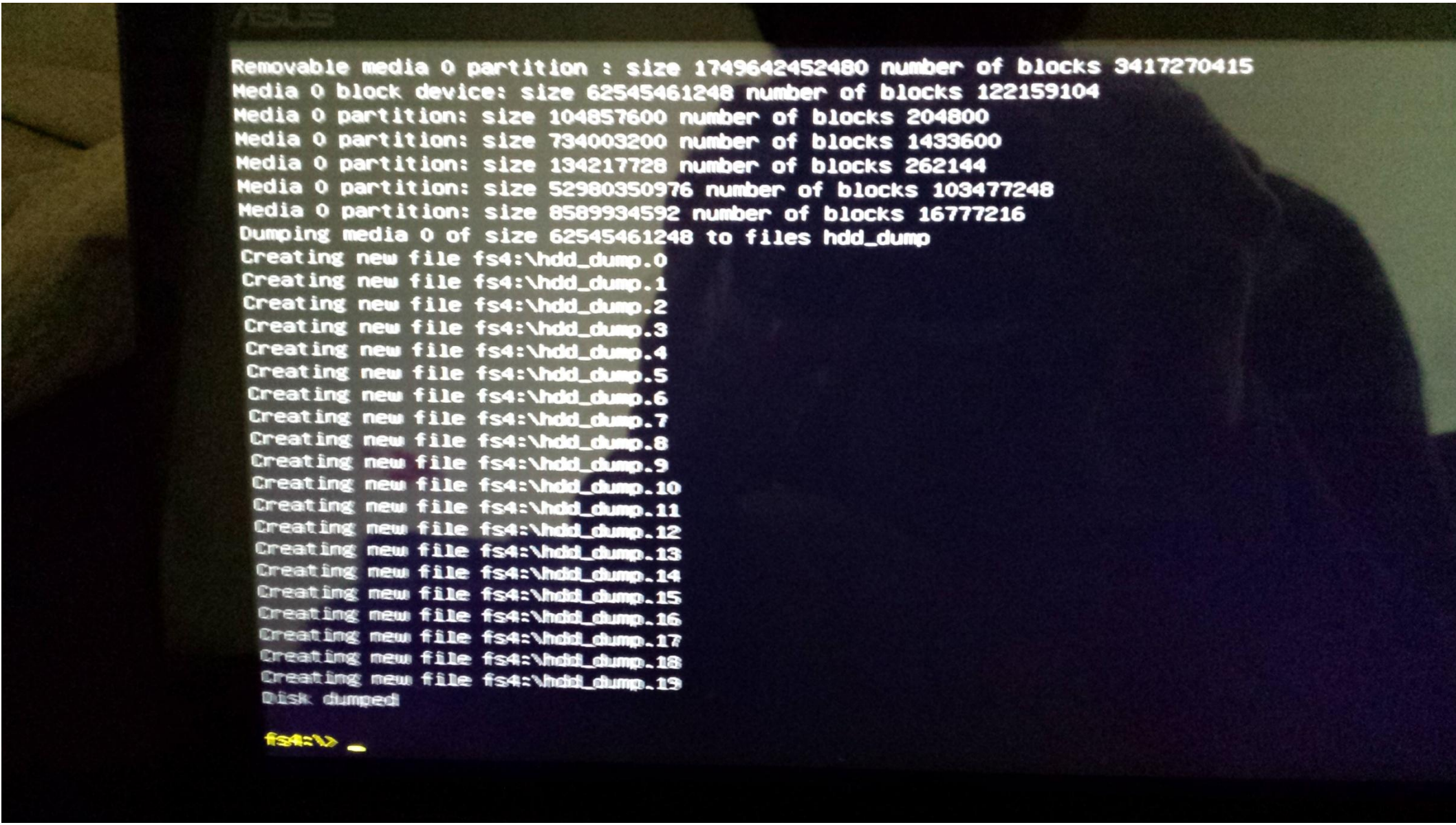
UEFI pour QEMU :

Très lent au redémarrage
Pas de données en RAM

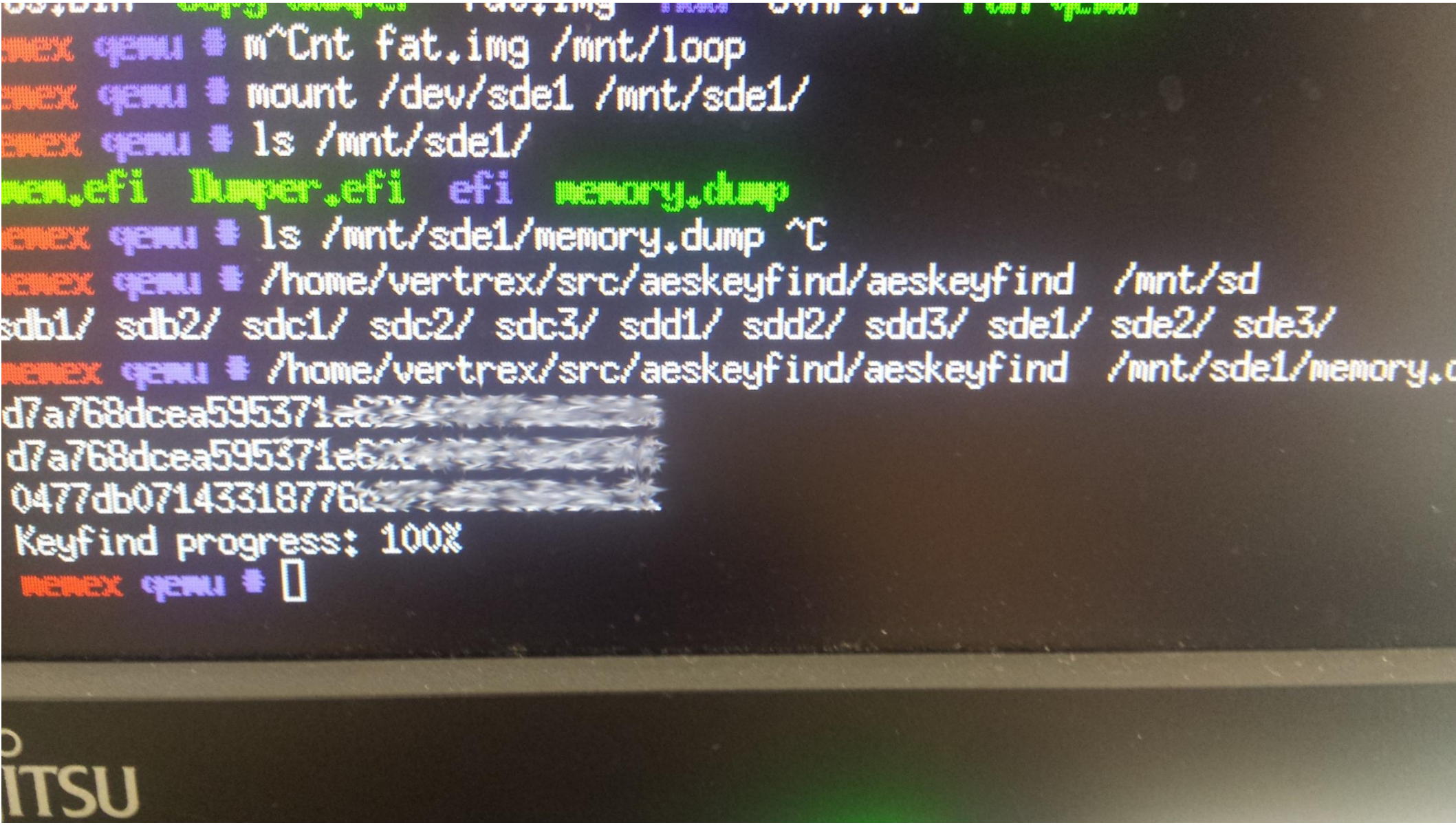
Résultat : Copie de la mémoire



Résultat : Copie du disque



Résultat : Extraction de la clef de chiffrement



Avantages

- Très simple d'utilisation ("Plug'n'play")
- Indépendant du système d'exploitation
- 32 & 64 Bits
- Ne nécessite pas d'identifiant, mot de passe
- Copie forensique
- Empreinte mémoire légère
- Ne nécessite pas de carte réseau
- Copie effectuée sur la clef ou le disque USB où se trouve l'exécutable
- Permet de récupérer des infos complémentaires pour l'analyse :
  - Sur les périphériques (Mémoire, disque, carte réseau, …)
  - Heure système
  - …

Améliorations futures

- Fournir un binaire précompilé
- Menu de sélection des différents choix
- Détection automatique des disques chiffrés
- Intégrer la sortie directement dans un logiciel forensique (ex: DFF)
- Recherche de la clef et déchiffrement automatique des disques

Questions