


 Université Blaise Pascal  
 (Université Clermont Auvergne)  
 © 2016 A. Sigayret

**ISIMA**

## Systèmes d'exploitation

### (3) Du côté matériel

CM3 - 2016 v.1 1

## 1. Structuration d'un S.E.

### Les deux extrêmes :

- Côté utilisateur :
  - + Interface  
(métaphores de la machine à écrire, du bureau, ...)
  - + Mécanisme d'abstraction du matériel  
(arborescence de fichiers, ...)
  - + Outils de traitement et de programmation  
(API multimedia, ...)
- Côté machine :
  - + Gestion des périphériques
    - ← pilotes
  - + Gestion des ressources matérielles  
(processeur, MEV, tâches (processus, threads,...))

### Communication cloisonnée :

- Utilisateur → S.E. : appels systèmes
- S.E. → Machine : interruptions

© 2016 - A. Sigayret 2

## 2. Processeur et S.E.

### Jeu d'instructions (processeur CPU + coprocesseurs)

- **RISC** : jeu réduit  
exp. ARM
- **CISC** : nombreuses instructions  
exp. Intel x86
  - + Instructions spécialisées ?
    - calculs en nombres réels / vectoriel
    - affichages ("accélération") 2D / 3D
    - codecs audio / vidéo
  - ← instructions matérielles ou logicielles

### Taille du "mot-mémoire" : 8 b / 16 b / 32 b / 64 b / ...

- taille des registres (unité mémoire)
- nombre d'instructions  $2^8 / 2^{16} / 2^{32} / 2^{64}$
- adressage MEV ou HD : taille du bloc / nombre de bloc

### Assembleur : jeu d'instructions processeur + fonctionnalités S.E.

- Interruption : "routine" préprogrammée pour {processeur, S.E.}  
exp interruption E/S :
  - MSDOS : int 21h
  - Linux : int 80h
- Structuration du fichier exécutable (et positionnement en MEV)  
← pour tous : données+code

© 2016 - A. Sigayret 3

## 2. Processeur et S.E.

### Structuration du fichier exécutable

#### Formats successifs pour Unix/Linux :

- **Assembler Output** ("a.out") : obsolète
- **ELF** (*Executable and Linkable*) :
  - entête : 7F, E L F ... (type 16/32/64 b) ...
  - table des (éventuels) segments (pour MEV)
  - table des sections (pour "liens" : cf bibliothèques)  
← man elf, file, readelf, objdump
  - sections : .data / .rodata / .text

gcc objet1.c ...

objet 1	...	objet n
---------	-----	---------

gcc -static ...

objet 1	...	objet n	bibliothèque 1
---------	-----	---------	----------------

© 2016 - A. Sigayret 4

## 2. Processeur et S.E.

### Structuration du fichier exécutable

#### Formats Microsoft :

- .COM : pour un unique segment mémoire (≤64 kiB)  
→ pour MSDOS initial
- PE (Portable Executable)  
→ versions successives (dont 32 b / 64 b)  
← rupture Windows NT  
→ importance des DLL
- .exe : entête "MZ" + segment DOS\* / entête PE + segments PE



\* exp : "This program must be run under Win64"

© 2016 – A. Sigayret

5

## 2. Processeur et S.E.

### Structuration du fichier exécutable

#### Format OS X : Mach-O

- non compatible Unix/Linux
- ← initialement binaire multiple (\**Universal Binary*\* : PowerPC + Intel)
- entête du fichier indiquant architecture (PowerPC x86, IA-64)

© 2016 – A. Sigayret

6

## 3. Carte mère et S.E.

**Programme d'amorce** (firmware=micrologiciel=microcode)  
← un (ou plusieurs) composants programmés en RAM/ROM/...

#### sur PC-Intel :

- BIOS (*Basic Input/Output System*)  
- ... (matériel) ...  
- reconnaissance du disque de démarrage  
- chargement en MEV du MBR et exécution
  - MBR : 1<sup>er</sup> cylindre, 1<sup>er</sup> tête (plateau), 1<sup>er</sup> secteur (512 kB)  
(contient la table de partition du disque)
- le MBR installe le S.E.  
→ multi-OS : chargeur d'amorçage dans le MBR (Lilo, Grub)
- ← Limites : HD 2<sup>41</sup> B (~2,2 TB), faible sécurité, axé MS
- UEFI (*Unified Extensible Firmware Interface*)  
- "mini-S.E."  
- secteur d'amorçage GPT (*GUID Partition Table*)  
- gestionnaire d'amorçage intégré  
- GUID (*Global Unique Identifier*) : 16 B pour chaque S.E. autorisé  
← programmable (shell), HD 2<sup>32</sup> B (~10<sup>22</sup> B), MS+Linux\*+Apple, GUID payant, "tatouage" (SecureBoot & signature numérique\*\*)

© 2016 – A. Sigayret

\* Idisk remplacé par gdisk \*\* cf projet CoreBoot

7

## 3. Carte mère et S.E.

### Gestionnaires d'amorçage

- Installé dans le MBR
- Choisit la partition contenant S.E. à utiliser  
→ *multiboot* possible
- **LILLO** (Linux Loader) : licence BSD  
Léger  
→ ELILO : version UEFI
- **GRUB** : licence GNU  
Reconfigurable
- **isoLinux** (€ projet SysLinux) :  
pour DVD amorçable
- **Boot Camp** : licence apple  
→ installation d'un Windows sur Mac

amorçage :  
une problématique de SE

© 2016 – A. Sigayret

8

## 4. Disque dur et S.E.

### Gestion des disques

- Partitionnement
- Montage / démontage
  - ← disque amovible (dont clé USB)

géré par :  
système de fichiers

### Gestion des secteurs

- Un fichier dans un ou plusieurs blocs (*cluster*) du disque dur
  - petits fichiers : perte de place
  - gros fichiers : risque de fragmentation
- Associer à chaque fichier ses positions
  - ← vision linéaire du disque

### Gestion de l'arborescence des fichiers

- noms et type (dossiers / fichiers / liens)
- "métadonnées" : taille, dates (création, modification, accès), droits accès (r/w/x), propriétaire(s), ...
- relations entre dossiers (inclusion) et fichiers (appartenance)

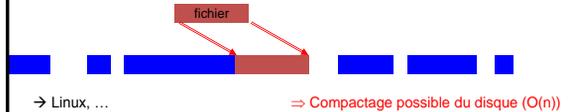
© 2016 - A. Sigayret

9

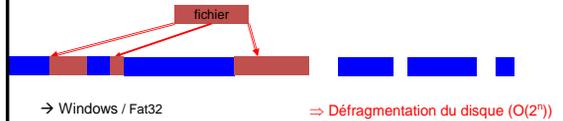
## 4. Disque dur et S.E.

### Exp Gestion de l'ajout de fichier

**1<sup>ère</sup> méthode** : insérer un fichier dans un bloc libre assez grand



**2<sup>ème</sup> méthode** : répartir le fichier dans les premiers blocs disponibles



© 2016 - A. Sigayret

10

## 4. Disque dur et S.E.

### Autres fonctionnalités possibles

- Lecture-écriture différées
  - ← cf imprimantes
- Journalisation des écritures
- Redondance multidisque (RAID)
- Partition multidisque
- Redimensionnement à chaud
- Compression
- Gestion des quotas
- + somme de contrôle, instantanés (*snapshots*), ...

© 2016 - A. Sigayret

11

## 4. Disque dur et S.E.

### Systèmes de fichiers (FS : File System)

→ principaux FS selon le S.E.

- Microsoft Windows
  - Fat32 (1996)
  - NTFS (1993)
- Linux
  - EXT3\* (1999)
- Apple OX X
  - HFS\*\* (1985) → HFS+
- Disques optiques (CD, DVD)
  - UDF (*Universal Disk format*, 1995)
    - ← RW : respecter norme ISO 9660 ?  
(cf Windows, Nero)
- + Portabilité inter-S.E. ?
  - clé USB : Fat32 ou NTFS
  - ← Samba pour ext3 sous Windows

© 2016 - A. Sigayret

\* accepté par BSD (natif : ZFS)  
\*\* accepté par ZIOS

12

## 4. Disque dur et S.E.

### Systèmes de fichiers : POSIX 1003.1

- Noms avec casse (Unicode)
- Permissions de traversée
- horodatage "heure de modification" (#DOS "heure de dernière modification")
- liens matériels ("durs", en plus des liens symboliques = "jonctions")

→ UNIX + LINUX + OS X + Win32/NTFS

## 4. Disque dur et S.E.

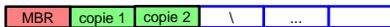
### Systèmes de fichiers : FAT32

- FAT : table d'allocation des fichiers,  
Table[cluster, état] ∈ {vide, réservé, utilisé+suivant, utilisé+dernier, erroné}  
Blocs de 4 à 64 KiB, partitions ≤ 2 TiB (théorique : 16 TiB)  
← limite fréquente de partition à 32 GiB
- adressage sur 28 b (4 b réservés) → ≤ ~2,7 10<sup>8</sup> fichiers (2<sup>28</sup>)  
(N.B. Fat16 : adressage sur 16 b)
- Nom de fichier ≤ 255 B en UTF-16 (sauf / \ ; \* ? \* < > | et espaces extrêmes)\*  
\* brevet logiciel jusq 2016 ?  
← héritage : "Program Files" = PROGRA~1 (système 8.3)
- Taille de fichier ≤ 4 GB (2<sup>32</sup>-1)
- Attributs de fichiers : archivé, caché, système, verrouillé
- Dates [01/01/1980..31/12/2107]
- + simplicité, interopérabilité  
- peu résistant, limites de taille, fragmentation

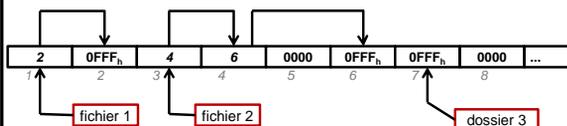
## 4. Disque dur et S.E.

### Systèmes de fichiers : FAT32

- Deux copies de la Table FAT



- Contiennent chaînes d'allocation des blocs



- Contiennent entrées de dossier/fichier
  - + Entrée courte : nom DOS (cf PROGRA~1) + méta-données
  - + Entrées longues : nom Windows, etc

## 4. Disque dur et S.E.

### Systèmes de fichiers : NTFS (≥v5)

- Arbre d'allocation **B-arbre** (binaire équilibré)  
Blocs de 1/2 à 64 KiB, partitions ≤ 256 TiB (théorique : 16 EiB)
- adressage sur 32 b → ≤ ~4 10<sup>9</sup> fichiers (2<sup>32</sup>-1)
- Nom de fichier ≤ 255 B en UTF-16 (sauf / \ ; \* ? \* < > | et espaces extrêmes)\*  
\* brevet logiciel jusq 2016 ?  
← bogue d'affichage si un changement maj/min
- Taille de fichier ≤ 16 TiB (théorique : ≤ 16 EiB)
- Attributs de fichiers : archivé, caché, système, verrouillé
- Dates [01/01/1601..28/05/60058]
- + journalisation, réparation, défragmentation, compression, indexation, cryptage, ...  
- propriétaire & peu documenté, faible interopérabilité (cf pilote NTFS-3G)

## 4. Disque dur et S.E.

### Systèmes de fichiers : NTFS

- Séparation des volumes d'un même disque
  - + Blocs logiques indépendants des blocs physiques
- Stockage des informations du volume dans des fichiers :
  - \$Mft (Master File Table), \$MftMirr (copie MFT), \$LogFile, \$Volume, \$, \$AttrDef, \$Boot, \$BadClus, \$Secure, \$UpCase, \$Extend
  - Noms des fichiers : > DOS, < POSIX
- Structure de la MFT : `fsutil.exe C: /info` (NTFS File Sector Information Utility. Copyright (C) Microsoft Corporation 1999. All rights reserved.)

```
File 0
Master File Table ($MFT)
STANDARD_INFORMATION (resident)
FILE_NAME (resident)
SDATA (nonresident)
logical sectors 6291456-6701311 (0x600000-0x66400f)
logical sectors 1033965800-103678887 (0x6293ca8-0x62e03a7)
BITMAP (nonresident)
logical sectors 12486032-12486135 (0xbae850-0xbae857)

File 1
Master File Table Mirror ($MFTMirr)
STANDARD_INFORMATION (resident)
FILE_NAME (resident)
SDATA (nonresident)
logical sectors 16-23 (0x10-0x17)

File 2
...
```

## 4. Disque dur et S.E.

### Arborescence de fichiers : Windows

```
[Bureau] ← avant "Modern UI"
[Ce PC]
Disque C:
  Program Files (x86)
  ProgramData
  Program[ne]s
    Windows Defender
    Windows NT
    WindowsPowerShell
    ...
  User
    nom
      Application Data
      Bureau
      Documents
      ...
  Windows
    System
    System32
    Temp
    ...
```

N.B. lien = fichier .lnk  
commande tree.com

- Héritages : avant NT / 32 b / 64 b
- Mélanges Système/Applications, Traitement/Données

## 4. Disque dur et S.E.

### Partitions : Windows

- Une seule partition
- Un fichier d'échange
- Fabriquants :
  - + partition de récupération, ...
  - Gestion de l'ordinateur / Stockage / gestion des disques



## 4. Disque dur et S.E.

### Systèmes de fichiers : EXT3

- Tables + Arbre
  - Blocs physiques de 1/2 à 64 kiB codés en 32b
  - partitions ≤2 TiB à ≤32 TiB
- adressage sur 16 b ou 64 b → ≤ 2 ou 32 Ti fichiers
- =nombre d'inode
- Nom de fichier ≤255 B en UTF-8 (sauf / 0)
- Taille de fichier ≤16 GiB ou ≤2 TiB
- Attributs de fichier : no-time, append-only, synchronous-write, no-dump, h-tree (directory), immutable, journal, secure-delete, top (directory), allow-undelete
  - + droit Unix {rwx}3
- Dates [14/12/1901..18/01/2038]
- + libre, chiffrement
- limite de date (→ EXT4, BTRFS,...)

## 4. Disque dur et S.E.

### Systemes de fichiers : EXT3

- Adressage logique :  $2^{32}$  blocs de 4kiB → 16TiB adressable (adressage d'un bloc logique sur 32 b)
- Adressage d'un fichier direct/indirect
  - direct :  $\leq 12$  blocs de 4 kiB → taille fichier  $\leq 48$  kiB
  - une indirection : 48 kiB < taille fichier  $\leq 4$  MiB
  - deux indirections : 4 MiB < taille fichier  $\leq 4$  GiB
  - trois indirections : 4 GiB < taille fichier
- au plus 32 000 sous-dossiers d'un dossier

## 4. Disque dur et S.E.

### Systemes de fichiers : EXT4

- Listes chaînées, arbre H, "extent" basé sur blocs logiques et non physiques → partitions  $\leq 1$  EiB
- adressage sur \*\*\* →  $\leq 4$  Gi fichiers → =nombre d'inode
- Nom de fichier  $\leq 255$  B en UTF-8 (sauf / 0)
- Taille de fichier  $\leq 16$  TiB
- Attributs de fichier : extents, noextents, mballloc, nomballloc, delalloc, nodelalloc, data=journal, data=ordered, data=writeback, commit=sync, orlov, cfballoc, user\_xattr, noouser\_xattr, acl, noacl, bsdif, minixdf, bh, nobh, journal\_dev + droit Unix {rwx}<sup>3</sup>
- Dates [14/12/1901..25/04/2515]
- + libre, chiffrement, journalisation, fiabilité, compatibilité ascendante
- journalisation, portabilité Win NT / OS X ?

## 4. Disque dur et S.E.

### Systemes de fichiers : EXT4

- Adressage logique :  $2^{48}$  blocs de 4kiB → 16TiB adressable (adressage d'un bloc logique sur 48 b)
- Adressage par "extent"
  - = portion de fichier sur des blocs consécutifs
  - 96 b = 48 b : adresse 1<sup>o</sup> bloc physique
  - + 15 b : taille extent
  - + 32 b : n<sup>o</sup> de bloc logique
- sous-dossiers d'un dossier :
  - au plus 64 000 ?
  - illimité ?

## 4. Disque dur et S.E.

### Gestion de fichiers : inoeuds sur Unix/Linux

- Un identifiant numérique pour chaque fichier / dossier / lien symbolique  
N.B. liens matériel pour fichiers d'une même partition
- Pointe sur blocs logique du fichier (adressage direct ou indirect)
- Avec chaque inoeud (inode) :
  - taille du fichier, périphérique conteneur, propriétaire (UserID, GroupID), {rwx}<sup>3</sup>, dates (modifications, accès), nombre de liens matériels vers cet inode (Nlinks).
- Inventaire d'un dossier : inoeuds
- Commandes : ls -li, ls -li, stat, stat -f, du (blocs occupés), df (blocs libres)

```
stat -f /cygdrive/c
Fichier : /cygdrive/c/*
Identif. : 40736480000000 Longueur du nom : 255 Type : UNKNOWN (0x07007)
Taille de bloc : 4096 Taille de bloc fondamentale : 4096
Blocs total : 111029143 libre : 96915576 disponible : 96915576
Inoeuds : total : 184467440/3709551615 libre : 1

stat /cygdrive/c
Fichier : /cygdrive/c/*
Taille : 0 Blocs : 8 Blocs d'Es : 65536 repertoire
Pereurheber : 40736480/3709551648 Inoeud : 140739488/51288 Liens : 1
Acces : (0070/3-rw-r--r--) UID : (128384/NT SERVICE+TrustedInstaller) GID : (128384/NT SERVICE+TrustedInstaller)
Acces : 2016-03-08 16:28:04.515709500 +0100
Modifie : 2016-03-08 16:28:04.515709500 +0100
Change : 2016-03-08 16:28:04.515709500 +0100
Cree : 2012-07-26 07:32:51.245230500 +0200
```

## 4. Disque dur et S.E.

### Arborescence de fichiers : **Unix/Linux** (≠POSIX)

```
/
...
/bin      commandes de base
/boot    chargeur d'amorçage
/dev     { hda, stdin, stdout, tty, ... }
/etc     configuration
/home    dossiers utilisateurs
/lib     {*-lib*} bibliothèques logicielles pour bin et/sbin
/media   montage (amovible)
/mnt     montage (temporaire)
/opt     programmes optionnels
/proc    "processus" (cf noyau)
/root    "home" du superutilisateur (S.U.)
/sbin    commandes du système (pour S.U.)
/tmp     fichiers temporaires
/usr     pour les utilisateurs
/var     { bin, lib, ... } données variables (liens). Exp /var/mail
...
```

→ Variantes selon systèmes. Exp /run, /vmlinuz

## 4. Disque dur et S.E.

### Partitions : **Unix/Linux**

- Un point de montage pour chaque partition d'un disque
- Partition d'échange (swap)
- Exp
  - /dev/hda1 – ext3 – point de montage : /
  - /dev/hda2 – ext3 – point de montage : /home
  - /dev/hda3 – swap – partition d'échange
- Mémorisation des points de montage dans /etc/fstab

## 4. Disque dur et S.E.

### Systemes de fichiers : **HFS+**

- B-arbres (contenu répertoire, mauvais blocs) + bitmap (allocation fichiers)  
Partitions ≤8 EiB
- adressage sur 32 b → ≤ ~4 10<sup>9</sup> (2<sup>32</sup>-1) fichiers
- Nom de fichier ≤255 B en variante Apple (*case-insensitive*) de **UTF-16**
- Taille de fichier ≤8 EiB
- Attributs de fichier :
  - couleur (3 bits), verrouillage, icônes personnalisable, archive, caché, alias, système, initrd, no INIT resources,
  - partage\_bureau
  - + droit Unix {rwx}<sup>3</sup>
- Dates ?[01/01/1904..06/02/2040]
- + compression, chiffrement
- propriétaire, pas de somme de contrôle, case-insensitive, limite de date

## 4. Disque dur et S.E.

### Systemes de fichiers : **UDF**

- Une norme de FS commune au CD/DVD : ISO 9660
- Un format "image" de CD/DVD : .iso
- Des extensions à la norme (RockRidge, Joliet, ...)
  - plus grande profondeur d'arborescence
  - noms de fichiers moins contraints
- Fonctionnalités supplémentaires ou différentes de S.E. / Applications
  - problèmes d'interopérabilité

## 4. Disque dur et S.E.

### Le cas des mémoires "flash" (USB / SSD)

- Structuration en blocs → mêmes FS que disques classiques
- L'écriture "use" le support
  - ne pas écrire toujours dans les mêmes blocs
- SSD : nombreuses écritures
  - système de répartition des écritures sur les blocs
    - ← compatibilités = carte mère + SSD + F.S. + S.E.
    - + TRIM

## 4. Disque dur et S.E.

### Systèmes de fichiers : Evolution !

- Arbres plutôt que table
- Adressage  $\leq 8$  EIB ← processeurs 64 b
- Compression, journalisation, chiffrement, redimensionnement à chaud, ...
- Problématiques propres à un type de support

## 5. Périphériques

### Les pilotes

#### - Exemple de pilotes intégré au S.E.

- Carte réseau – protocole TCP/IP
  - pilote Windows
  - pilote Linux
  - ...
- Composant carte réseau "incomplet" + pilote intégré à Windows
  - pas sous Linux

#### - Exemple de matériel prévu pour un S.E.

- Imprimante USB+Wifi
  - pilote Windows USB+Wifi + fonctionnalités (niveau d'encre, ...)
  - pilote OS X USB+Wifi
  - pilote Linux USB

→ Standardisation connexionnelle mais ...

## 5. Périphériques

### Les écrans

VGA (640x480) – XGA (768x1024) – UXGA (1600x1200) – QXGA (2048x1536) – ...  
HDTV (1280x720) – FHDTV (1920x1080) – 4K2K (4096x216) – ...

→ beaucoup de pixel à gérer

- ← carte graphique
- ← S.E.

24 i/s – 25 i/s – 30 i/s – ...  
50 Hz – 60 Hz – ...

→ afficher rapidement (+synchronicité)

- ← carte graphique
- ← S.E.

### Les écrans tactiles

- 2D / 3D ?
  - coprocesseur "tactile" ?
  - S.E. : écran devient périphérique entrée-sortie

### Multiplés écrans

- en parallèle (vidéoprojecteur) → différences de format d'écran
- en série → continuité entre écrans

## 6. Mémoire(s)

### Vitesse et conservation

	vitesse*	TAM*
Registre du CPU	8000 MB/s	0,0005 ms
Cache 1 du CPU	...	
Cache 2 du CPU	...	
PCI express (carte graphique)	2000 MB/s	
Mémoire vive	1000 MB/s	0,05 ms
disque SATA-III SSD	600 MB/s	0,1 ms
USB 3.0	500 MB/s	3 ms
disque SATA-I HD	150 MB/s	
USB 2	60 MB/s	
CD 40x	6 MB/s	40 ms
DVD	1,5 MB/s	140 ms

\* valeurs indicatives

→ Le S.E. doit assumer (mémoires tampons)

© 2016 – A. Sigayret

33

## 6. Mémoire(s)

### La mémoire vive des Intel x86

- X86 – le précurseur : Intel 8008
  - registres 8 b / adressage 14 b hors CPU
- X86-16 – l'ancêtre : Intel 8086 (1978)
  - registres 16 b / adressage 20 b (circuit MMU du CPU)
  - mode segmentée : bloc 64 KiB, adressage offset+segment 1 MiB
- X86-16 – la transition : Intel 80286
  - registres 16 b / adressage 24 b (circuit MMU)
  - mode segmentée : bloc 64 KiB, adressage offset+segment 16 MiB
  - + mode "protégé" :
    - adressage mémoire linéaire
    - séparation des processus → sécurisation
    - accepte mémoire virtuelle (*swap*) → "pagination"
    - niveau de privilège d'accès mémoire → protection
- X86-32 – la rupture : Intel 80386 / 80486 / Pentium
  - registres 32 b / adressage direct 32 b (par CPU) → 4 GiB
  - ← hérite des modes segmentés et protégés
- X86-64 – les temps modernes (IA-64)
  - registres 64 b, adressage direct 64 b → 16 EiB ?!
  - ← plus d'adressage segmenté.

© 2016 – A. Sigayret

34

## 6. Mémoire(s)

### Gérer la mémoire vive

- Linéariser l'adressage mémoire si le processeur ne le fait pas
  - ← assurer l'héritage Intel
  - ← limites de taille (cf disques durs)
- Réserver un espace mémoire pour chaque processus :
  - délimiter, structurer, protéger, hiérarchiser les privilèges
- Chargement d'un exécutable en mémoire
  - principe de structuration général
  - détail de structuration selon S.E.



© 2016 – A. Sigayret

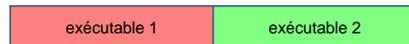
35

## 6. Mémoire(s)

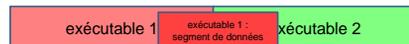
### Gérer la mémoire vive

#### Exemple de problème à gérer

1°) deux exécutables contigus en MEV



2°) écriture excédentaire dans un segment de données



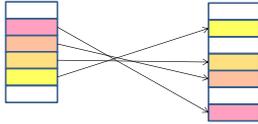
← fonctions problématiques du C (gets, ...)

© 2016 – A. Sigayret

36

## 7. Au-delà du matériel

- Virtualisation de la MEV  
adressage continu  $\leftrightarrow$  discontinuité matérielle



- Couche d'abstraction du matériel (Windows : HAL)  
commandes génériques  $\leftrightarrow$  pilote spécifique



- $\leftarrow$  problématiques communes  
MEV – disques durs, ...

FIN 37