

Le voyage d'un code

LANGAGES, BIBLIOTHÈQUES ET EXÉCUTABLE



Le voyage d'un code

SOMMAIRE

- I Compilation (traduction)
- II Edition des liens, construction
- III Chargement et exécution
- IV Mémoire virtuelle, mémoire physique
- V Quelles interfaces de code à code ?
- ANNEXES



Programmeuse

Code



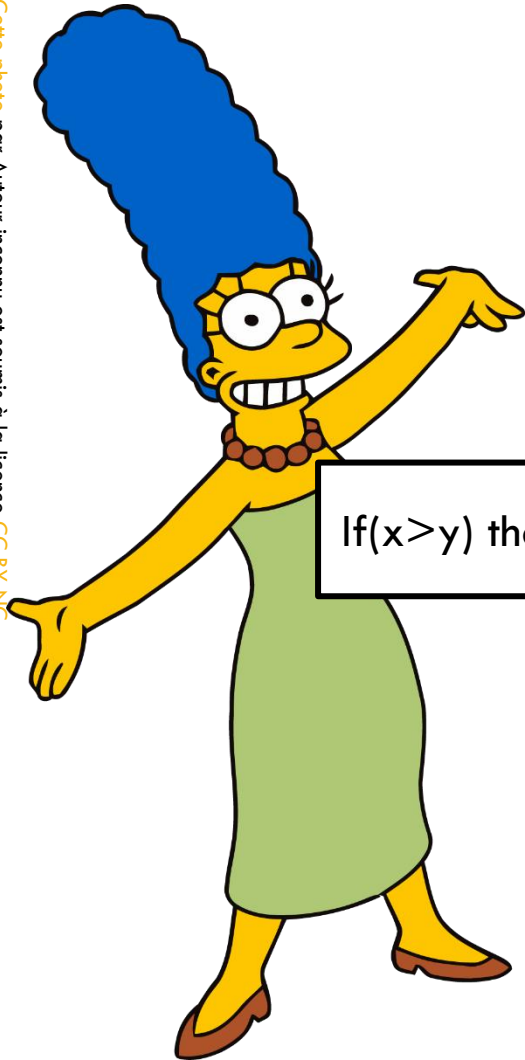
La machine veut son langage

I – Compilation (traduction)

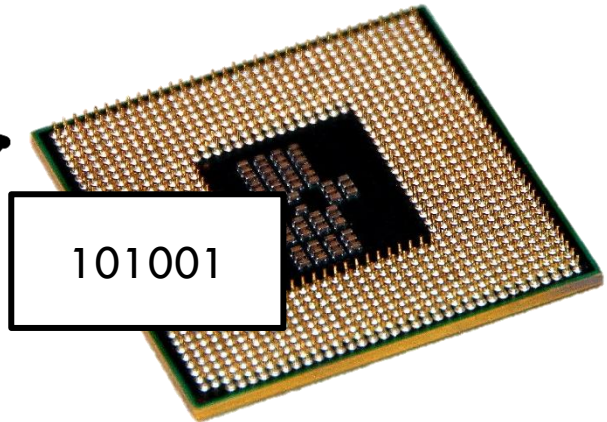
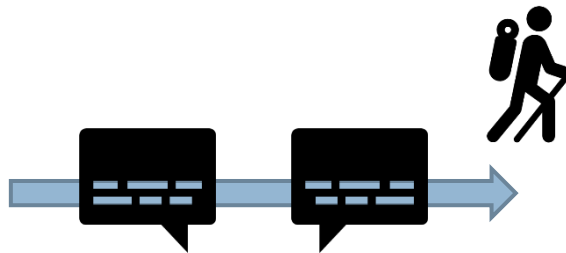
LA COMPILATION TRADUIT LE LANGAGE DE PROGRAMMATION EN LANGAGE MACHINE.

ELLE PRODUIT UN FICHIER OBJET QUI CONTIENT LE CODE MACHINE ET DES MÉTADONNÉES POUR LA SUITE DE LA CONSTRUCTION.

1) Compilation (traduction)

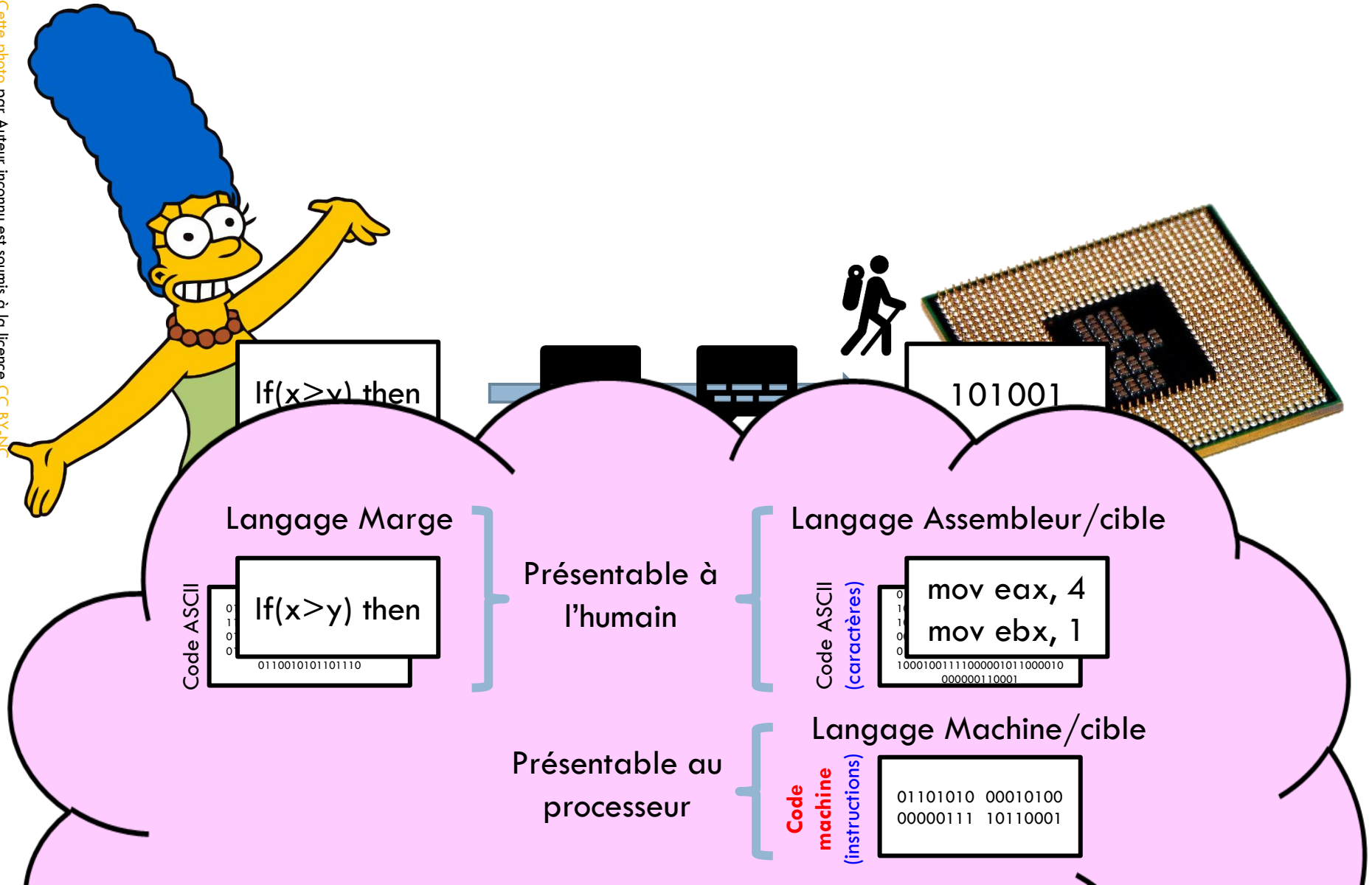


`If(x>y) then`

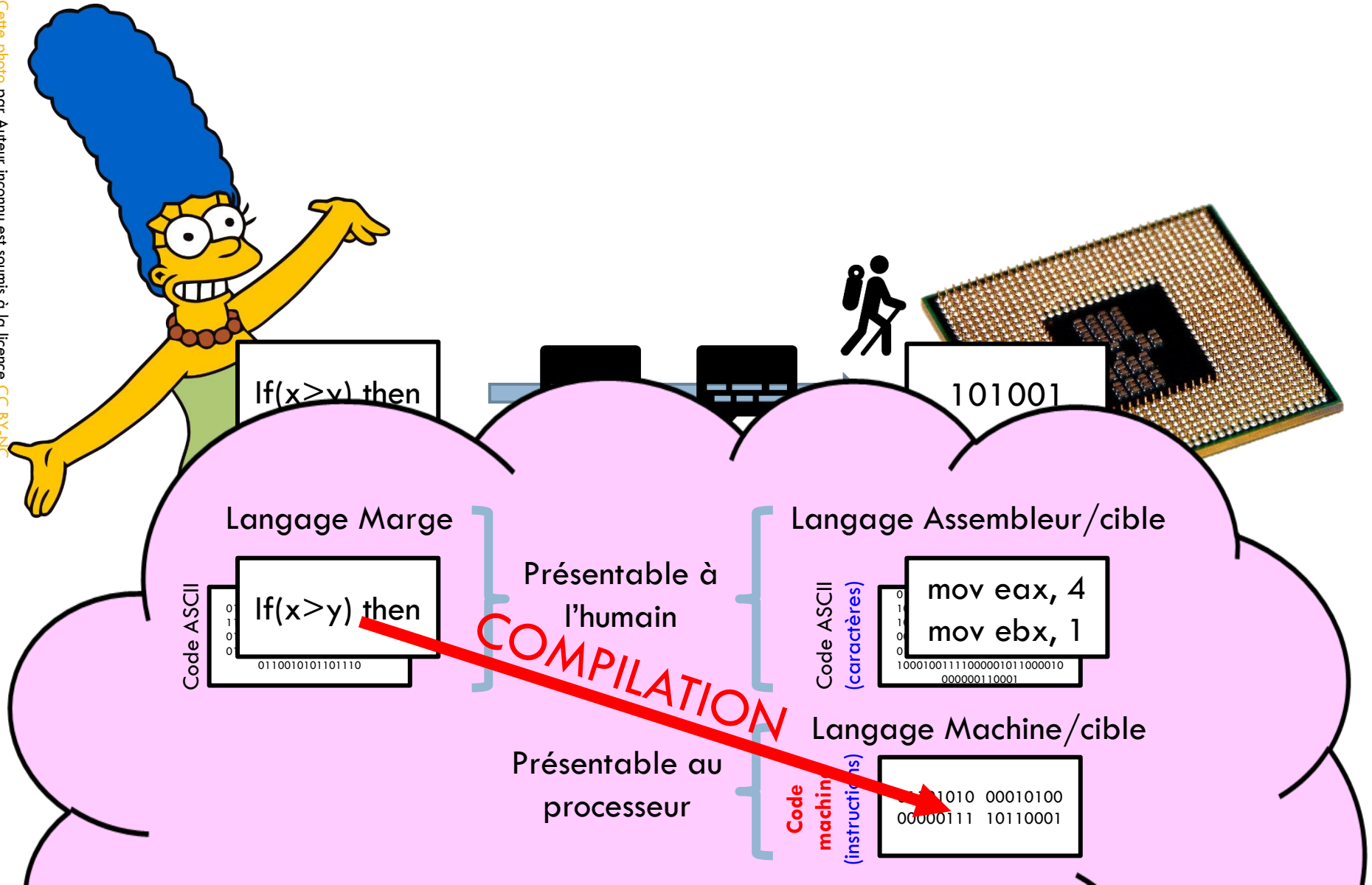


`101001`

1) Compilation (traduction)



1) Compilation (traduction)



1) Compilation (traduction)

OPCODE MAP

Exemple de Code machine

<https://software.intel.com/en-us/download/intel-64-and-ia-32-architectures-sdm-combined-volumes-1-2a-2b-2c-2d-3a-3b-3c-3d-and-4>

Table A-2. One-byte Opcode Map: (08H – FFH) *

	8	9	A	B	C	D	E	F
0	Eb, Gb	Ev, Gv	Gb, Eb	Gv, Ev	AL, Ib	rAX, Iz	PUSH CS ⁶⁴	2-byte escape (Table A-3)
1	Eb, Gb	Ev, Gv	Gb, Eb	Gv, Ev	AL, Ib	rAX, Iz	PUSH DS ⁶⁴	POP DS ⁶⁴
2	Eb, Gb	Ev, Gv	Gb, Eb	Gv, Ev	AL, Ib	rAX, Iz		
3	Eb, Gb	Ev, Gv	Gb, Eb	Gv, Ev	AL, Ib	rAX, Iz		
4	eAX REX.W	eCX REX.WB	eDX REX.WX	eBX REX.WXB	eRAX REX.RB			
5	rAX/r8	rCX/r9	rDX/r10	rBX/r11	rSI			
6	PUSH ⁶⁴ Iz	IMUL Gv, Ev, Iz	PUSH ⁶⁴ Ib	IMUL Gv, Ev, Ib	IN Yb			
7	S	NS	P/PE	NP/PO	L/I			
8	Eb, Gb	Ev, Gv	Gb, Eb	Gv, Ev				
9	CBW/ CWDE/ CDQE	CWD/ CDQ/ CQO	far CALL ⁱ⁶⁴ Ap	FWAIT/ WAIT	PUSHF			
A	AL, Ib	rAX, Iz	STOS/B Yb, AL	STOS/W/D/Q Yv, rAX	LO AL			
B	rAX/r8, Iv	rCX/r9, Iv	rDX/r10, Iv	rBX/r11, Iv	rSP/r12, Iv	rBP/r13, Iv	rSI/r14, Iv	rDI/r15, Iv
C	ENTER lw, Ib	LEAVE ^{d64}	far RET lw	far RET	INT3	INT Ib	INTO ⁱ⁶⁴	IRET/D/Q
D	ESC (Escape to coprocessor instruction set)							
E	near CALL ⁱ⁶⁴ Jz	near ⁱ⁶⁴ Jz	JMP far ⁱ⁶⁴ Ap	short ⁱ⁶⁴ Jb	AL, DX	eAX, DX	DX, AL	DX, eAX
F	CLC	STC	CLI	STI	CLD	STD	INC/DEC Grp 4 ^{1A}	INC/DEC Grp 5 ^{1A}

NOTES:

* All blanks in all opcode maps are reserved and must not be used. Do not depend on the operation of undefined or reserved locations.

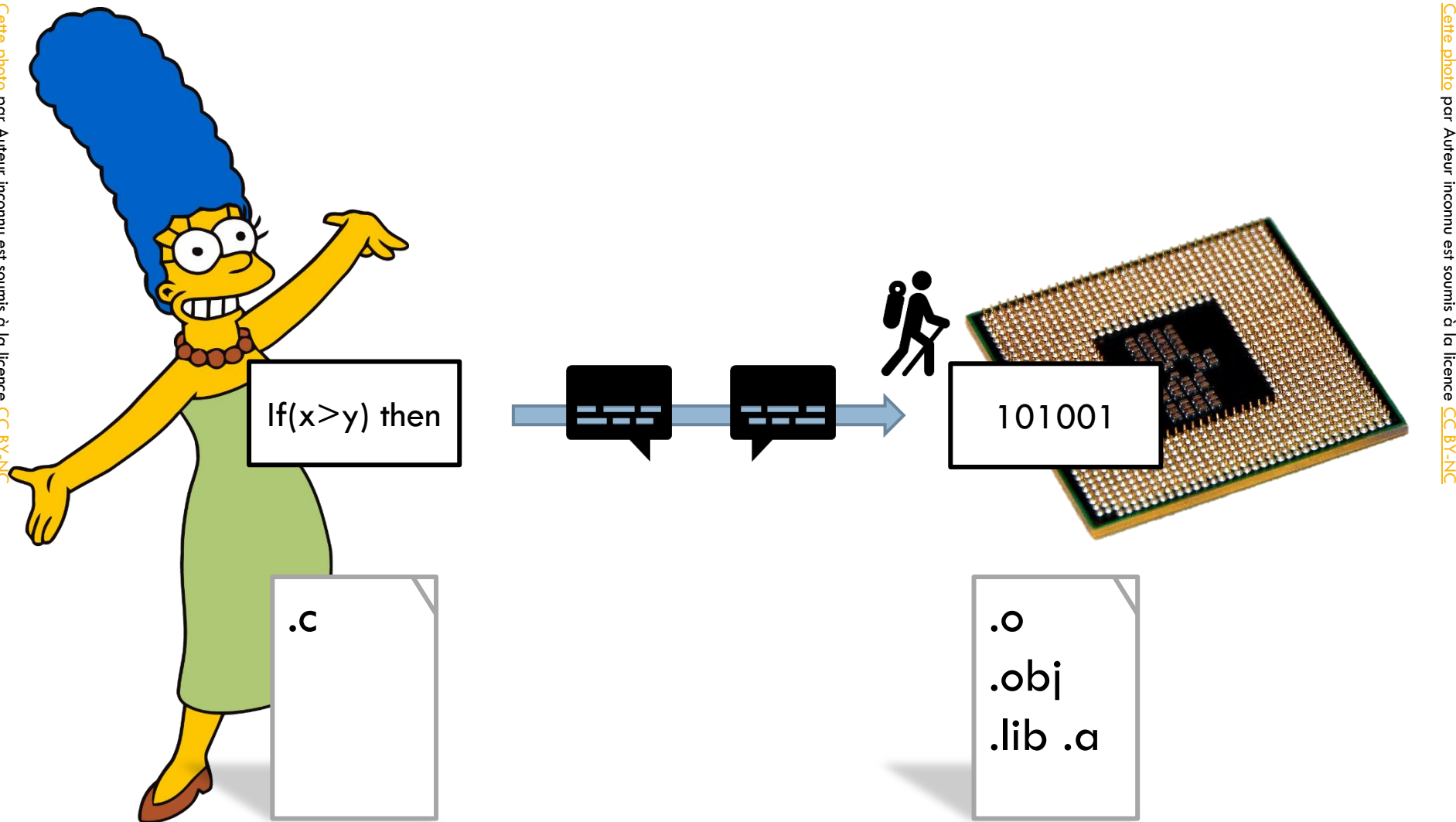


Intel® 64 and IA-32 Architectures Software Developer's Manual

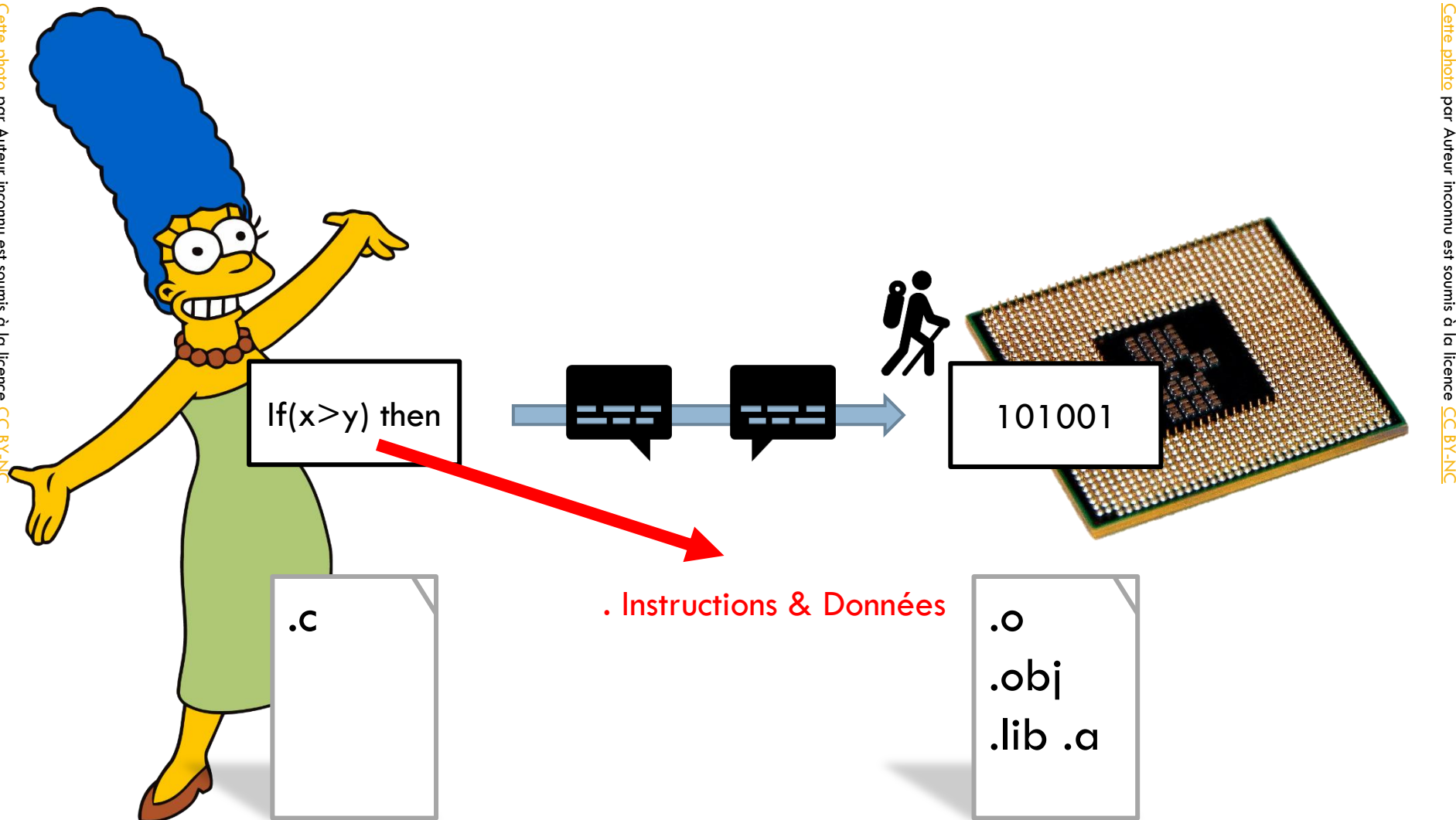
Combined Volumes:
1, 2A, 2B, 2C, 2D, 3A, 3B, 3C, 3D and 4

NOTE: This document contains all four volumes of the Intel 64 and IA-32 Architectures Software Developer's Manual: *Basic Architecture*, Order Number 253665; *Instruction Set Reference A-Z*, Order Number 325383; *System Programming Guide*, Order Number 325384; *Model-Specific Registers*, Order Number 335592. Refer to all four volumes when evaluating your design needs.

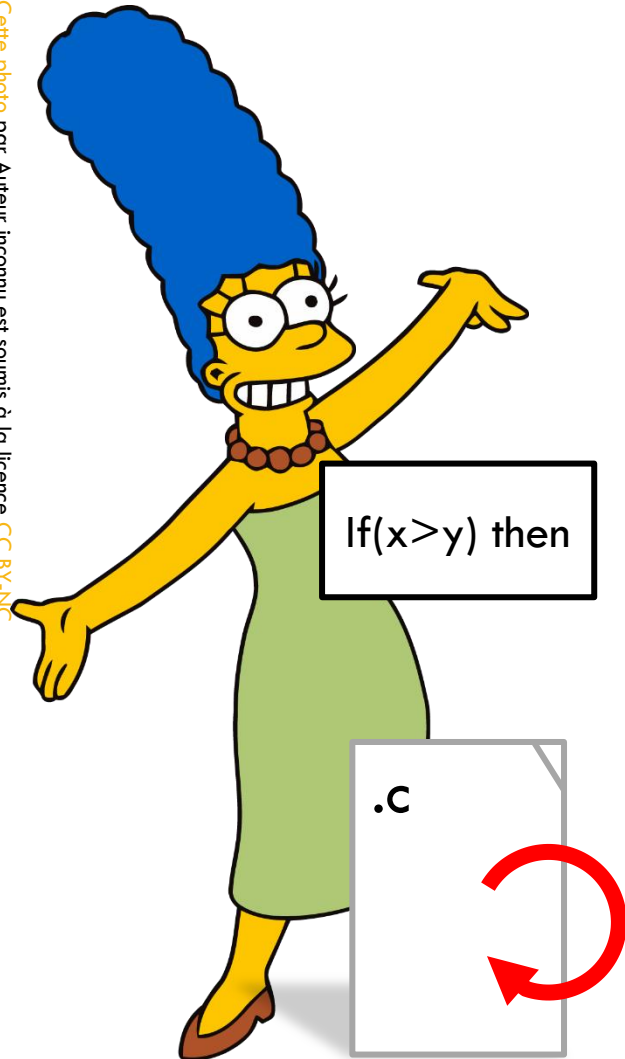
1) Compilation (traduction)



1) Compilation (traduction)

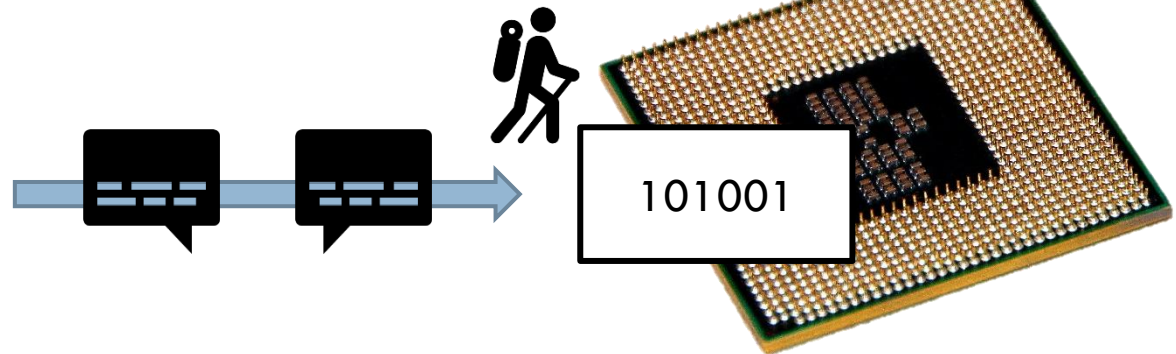


1) Compilation (traduction)



If(x>y) then

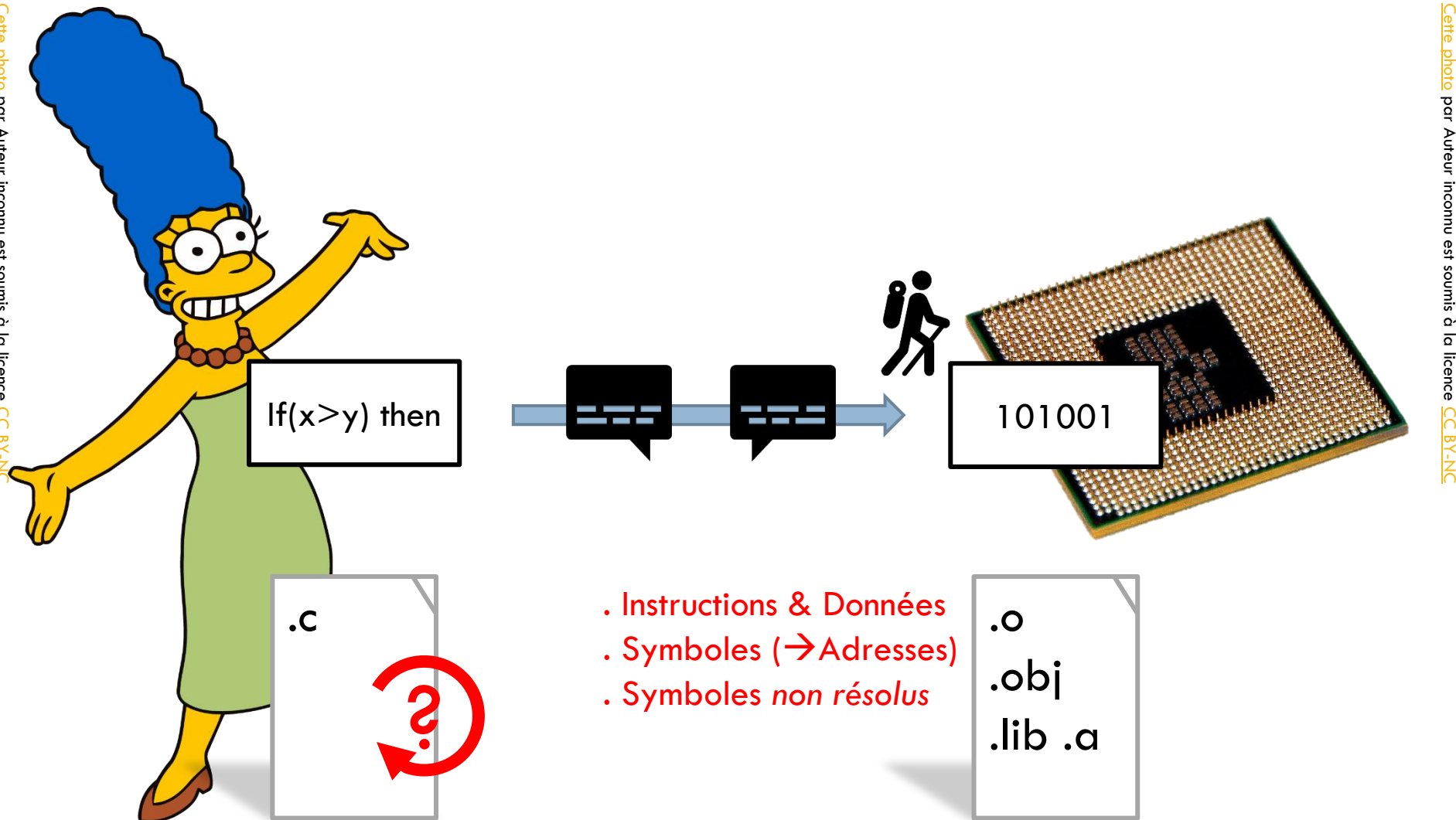
.c



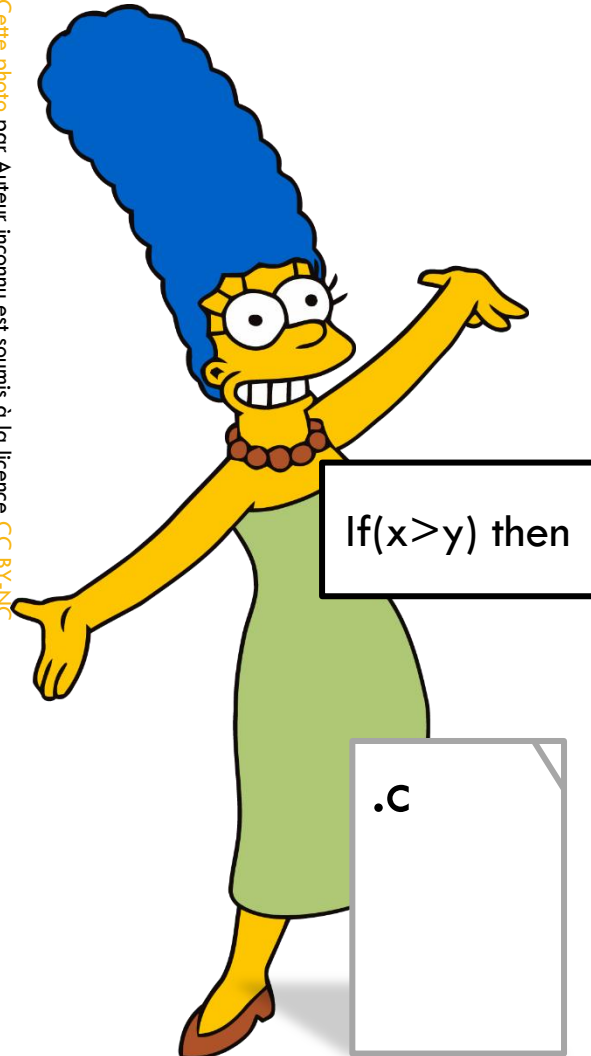
. Instructions & Données
. Symboles (→ Adresses)

.o
.obj
.lib .a

1) Compilation (traduction)

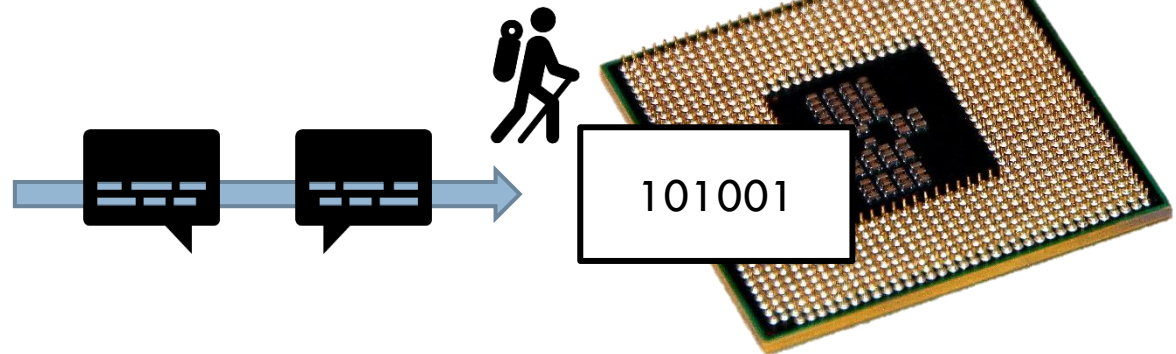


1) Compilation (traduction)



If(x>y) then

.c

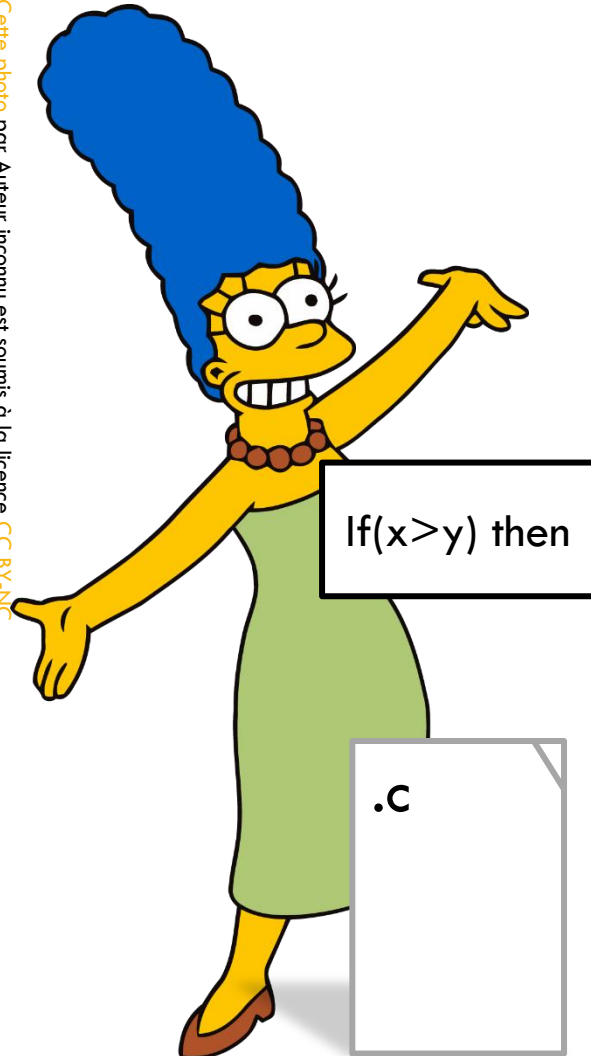


101001

- . Instructions & Données
- . Symboles (→ Adresses)
- . Symboles non résolus
- . Table de relocalisation

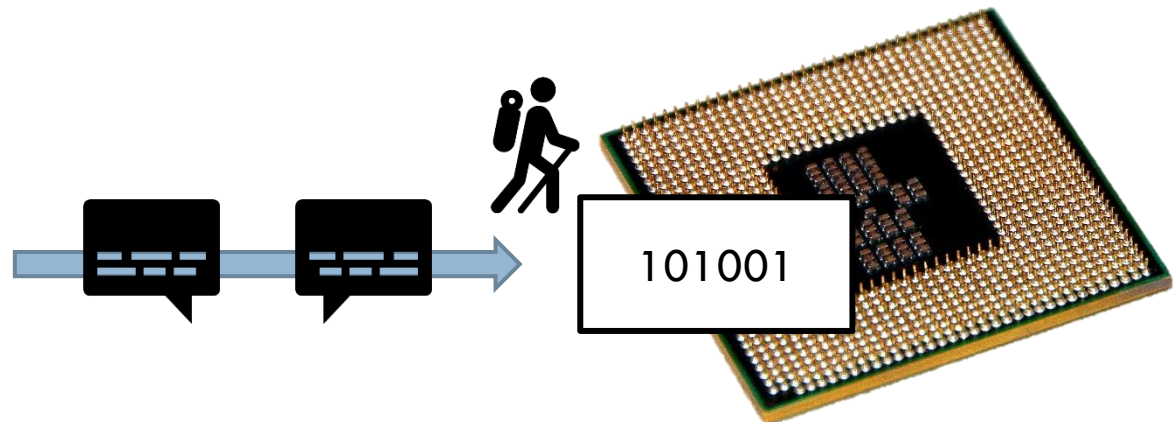
.o
.obj
.lib .a

1) Compilation (traduction)



If(x>y) then

.c



101001

- . Instructions & Données
- . Symboles (→ Adresses)
- . Symboles non résolus
- . Table de relocalisation
- . Noms décorés (*mangling*)

.o
.obj
.lib .a

1) Compilation (traduction)

La fameuse "décoration des noms"
de fonctions (*name mangling*)
[ici par un compilateur C++]

Appeler une fonction d'une DLL

Fonction Paramètres Callbacks Détection d'erreur

Nom ou chemin de la bibliothèque

C:\Users\marpaud\Desktop\cpp-function\
Calc_no_extern_c.dll

☐ Spécifier le chemin sur le diagramme

Nom de la fonction

Z14multiplicationdd

Z14multiplicationdd
Z8additiondd

Thread

☒ Exécuter dans le thread IU
☐ Exécuter dans n'importe quel thread

Convention d'appel

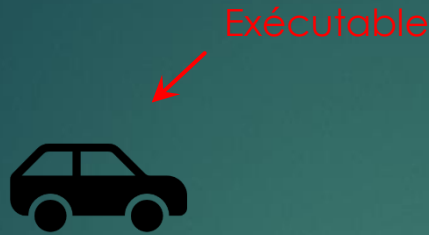
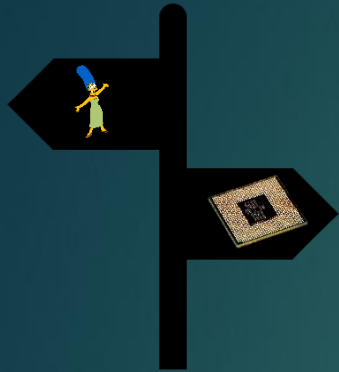
☐ stdcall (WINAPI)
☒ C

Prototype de la fonction

double _Z14multiplicationdd(double a, double b);

[Utilisez plutôt un assistant...](#)

OK Annuler Aide



Exécutable



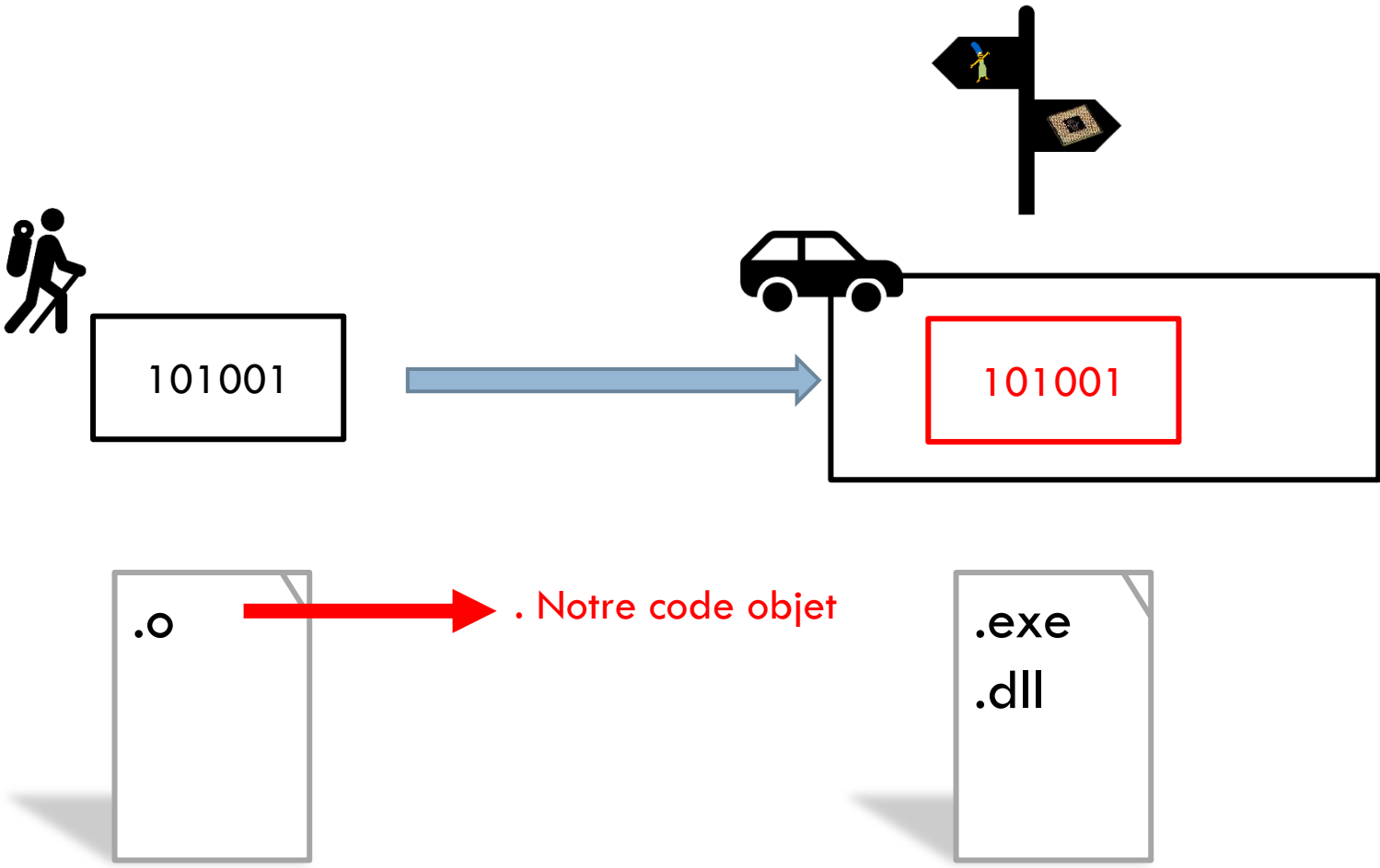
Code objet bogué prêt à intégrer son exécutable

II – Edition des liens, construction

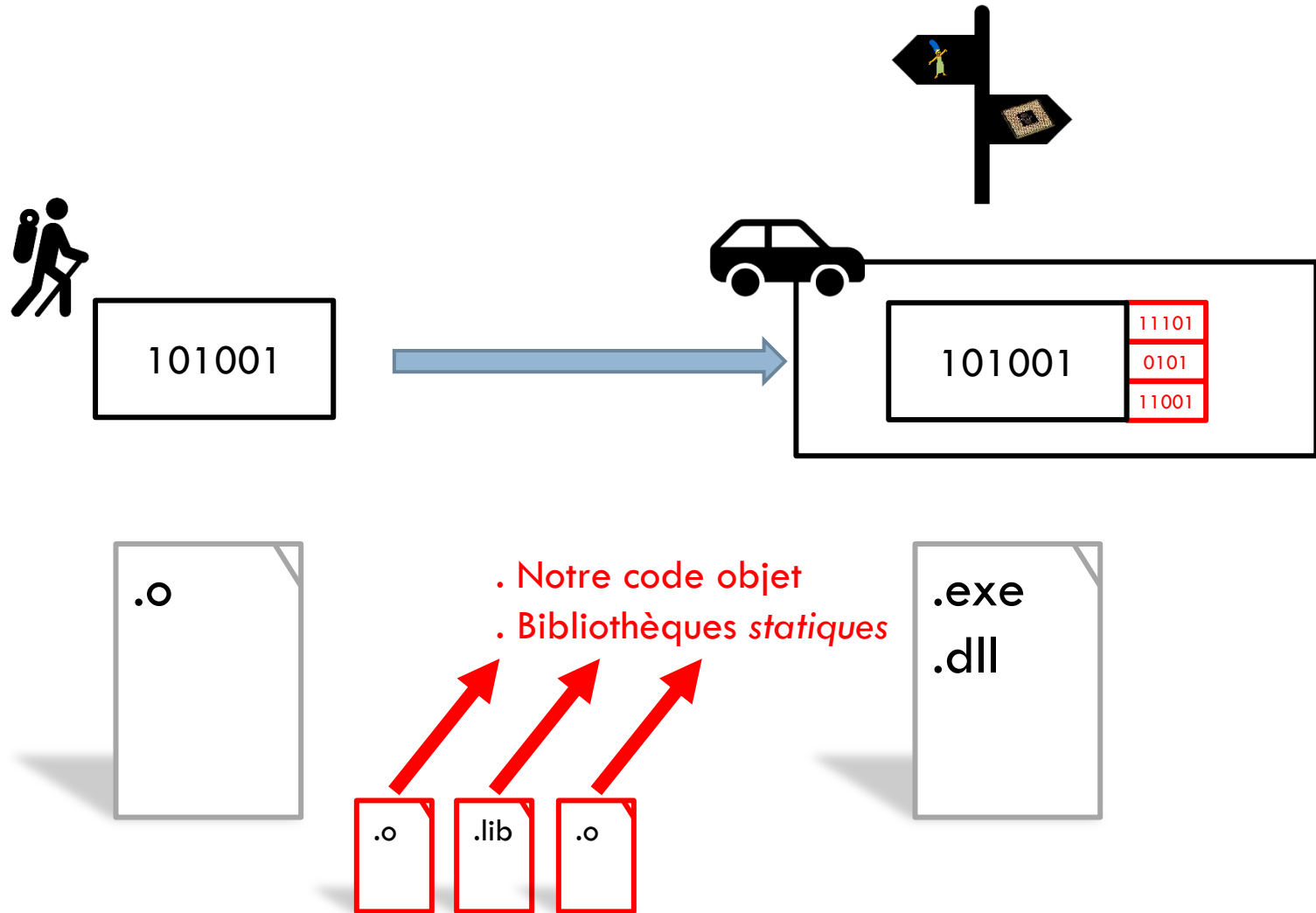
L'ÉDITEUR DE LIENS DU COMPILATEUR INCORPORE (LIENS STATIQUES) LE CODE MACHINE RÉFÉRENCÉ PAR LES SYMBOLES NON ENCORE RÉSOLUS.

LA CONSTRUCTION MET EN PLACE TOUTES LES INFORMATIONS REQUISES PAR LE FORMAT D'EXÉCUTABLE DE L'OS CIBLE.

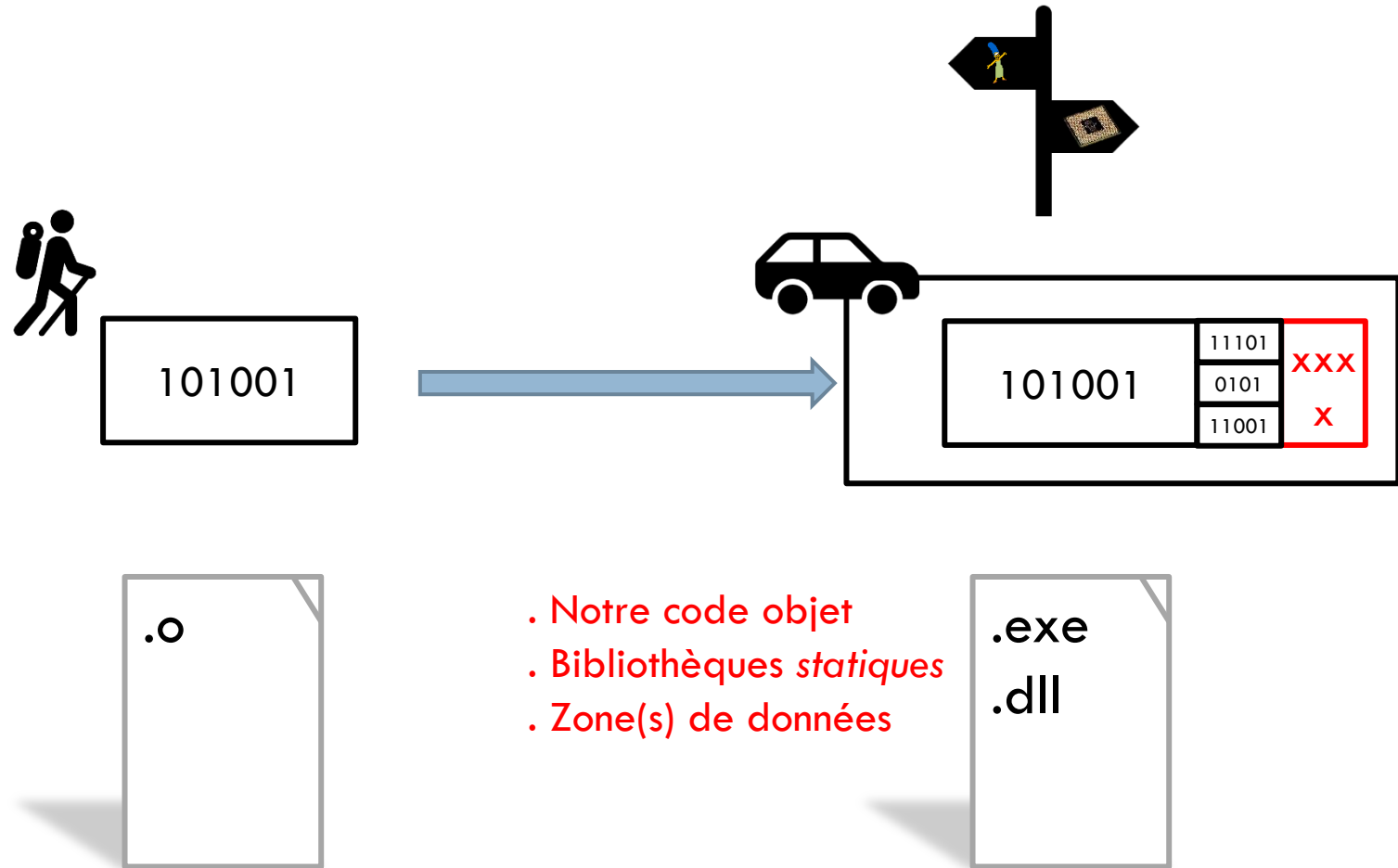
2) Edition des liens, construction



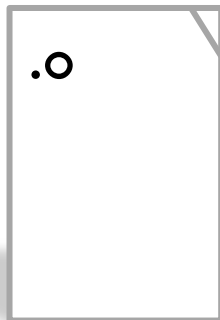
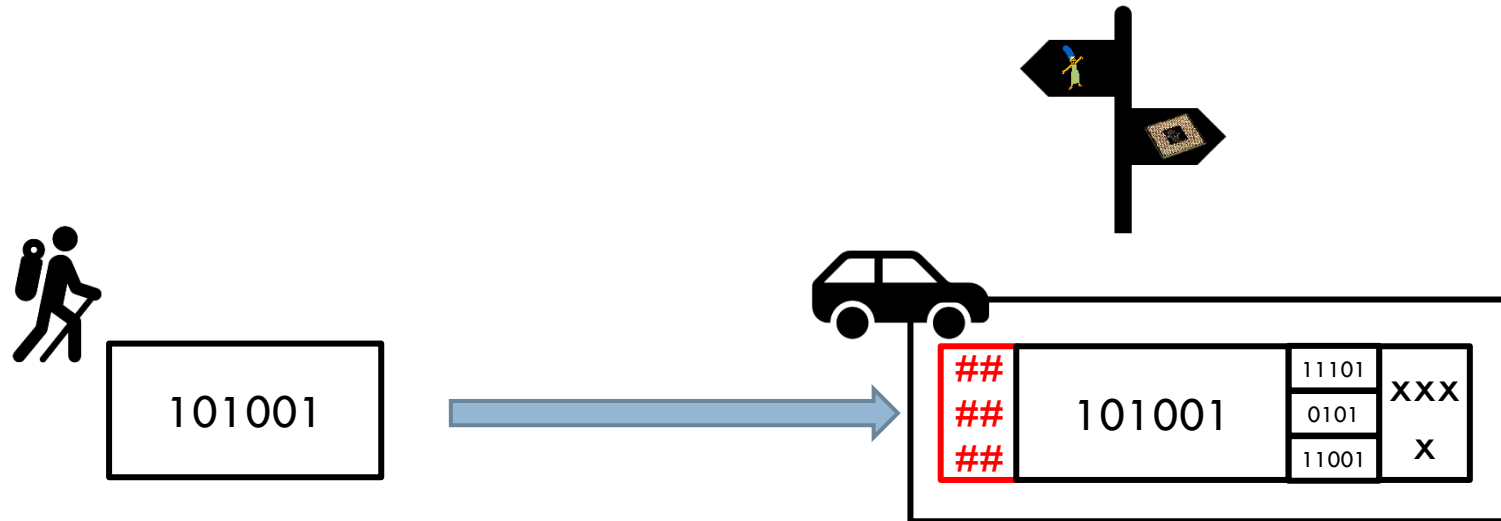
2) Edition des liens, construction



2) Edition des liens, construction



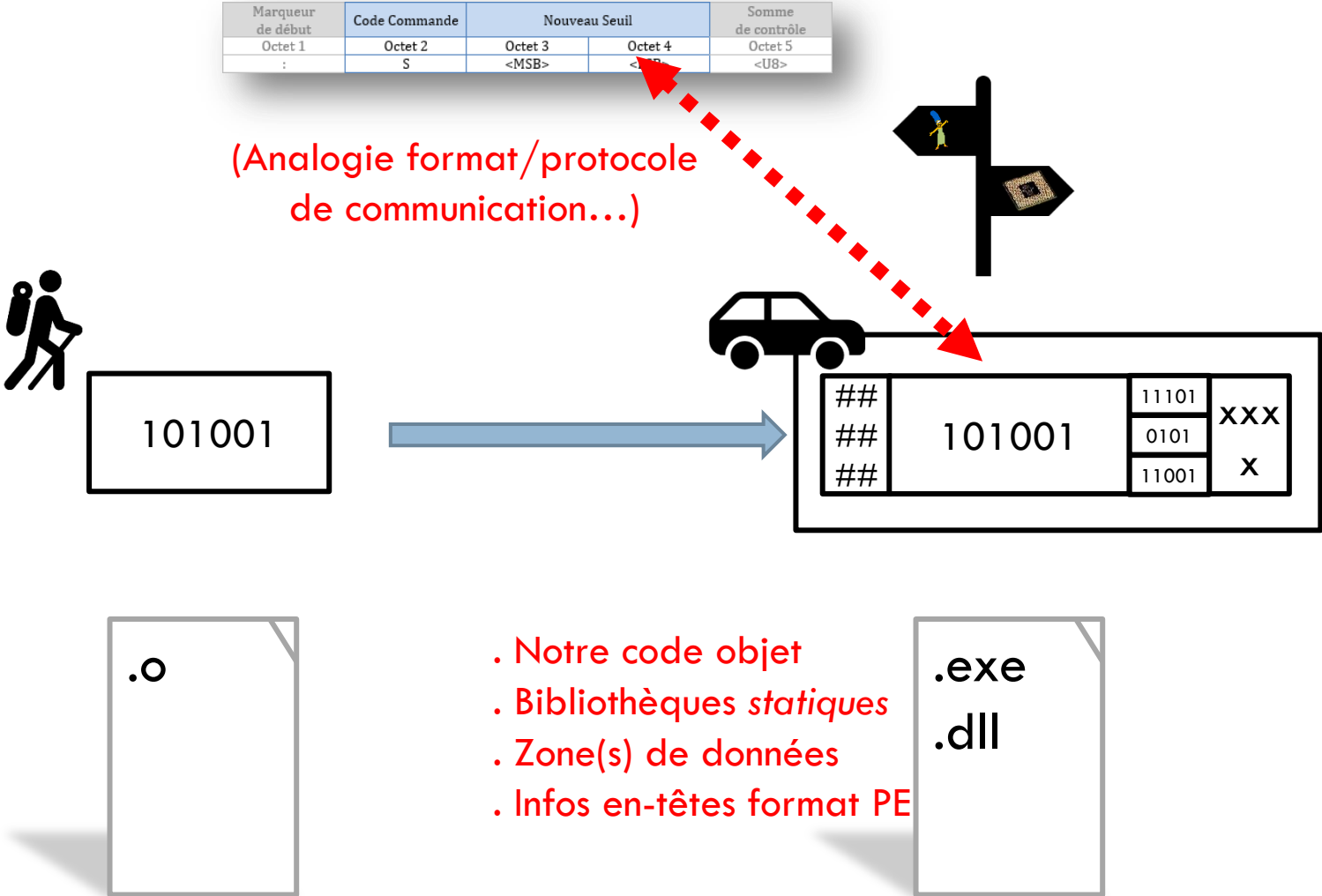
2) Edition des liens, construction

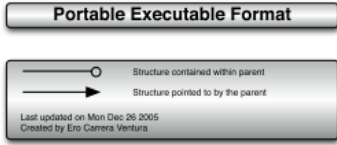


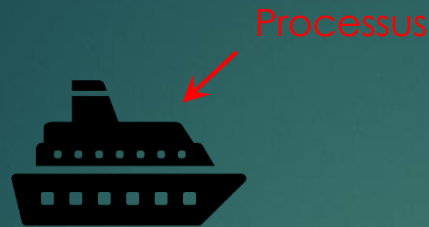
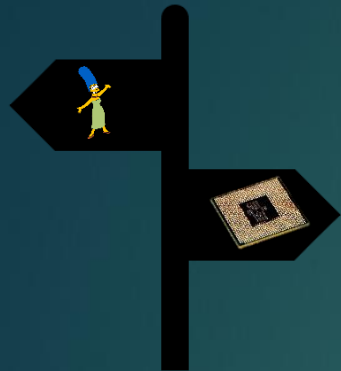
- . Notre code objet
- . Bibliothèques *statiques*
- . Zone(s) de données
- . Infos en-têtes format PE



2) Edition des liens, construction







Le processus est à quai

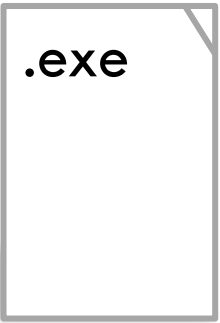
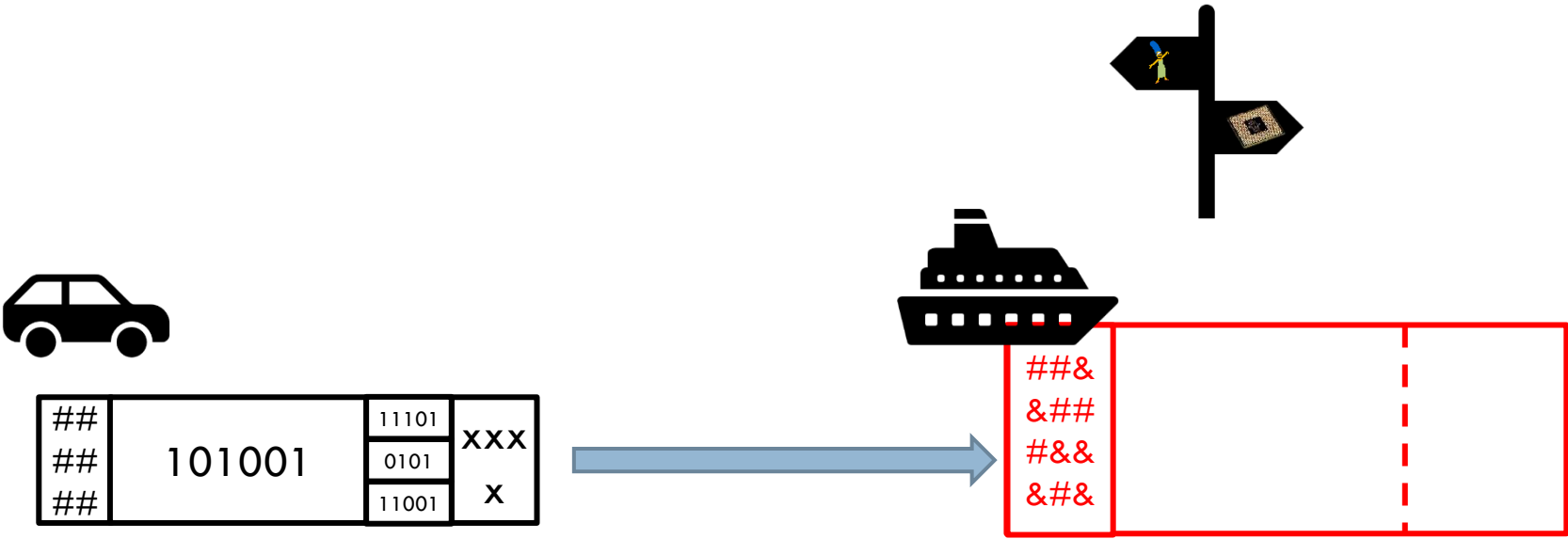
III – Chargement et exécution

LE LOADER (CHARGEUR) DE L'OS PLACE EN MÉMOIRE TOUT OU PARTIE DU FICHIER EXÉCUTABLE EN S'APPUYANT SUR LES MÉTADONNÉES INCLUSES.

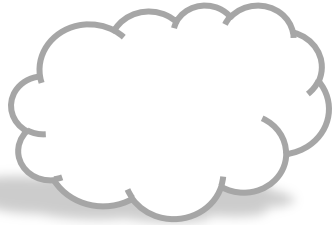
IL CHARGE OU RÉFÉRENCE LES BIBLIOTHÈQUES DYNAMIQUES.

L'ENSEMBLE INTÈGRE UN *PROCESSUS* LIÉ AUX RESSOURCES SYSTÈME.

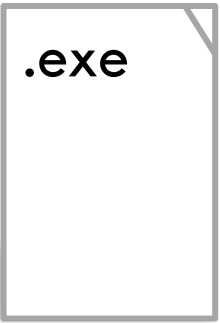
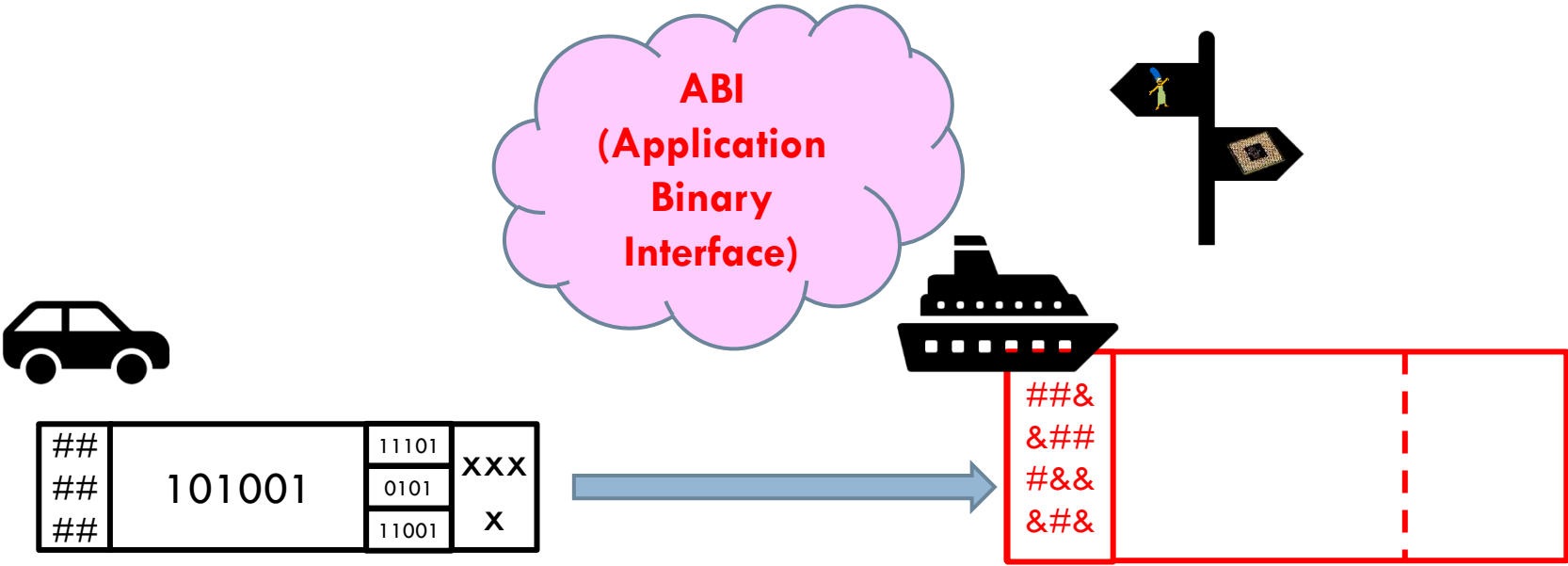
3) Chargement (processus en mémoire vive)



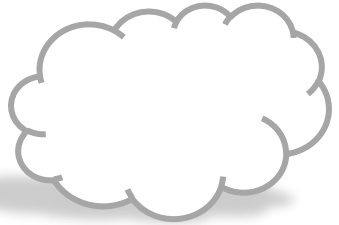
. Processus en RAM : zones de code, zones de données, pile..., 4Gio d'adressage virtuel



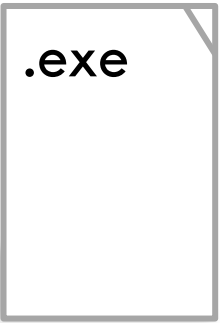
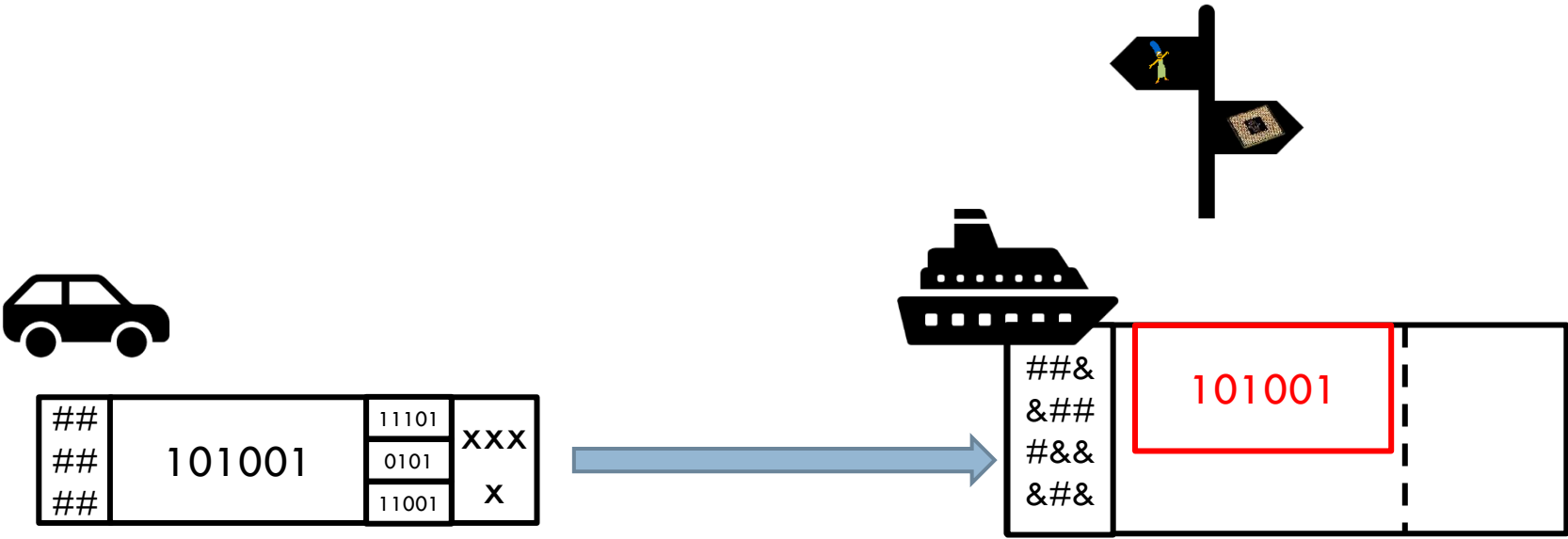
3) Chargement (processus en mémoire vive)



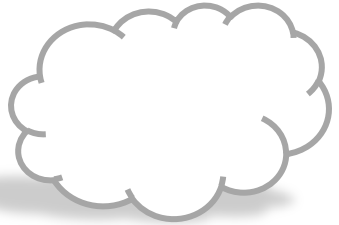
. Processus en RAM : zones de code, zones de données, pile..., 4Gio d'adressage virtuel



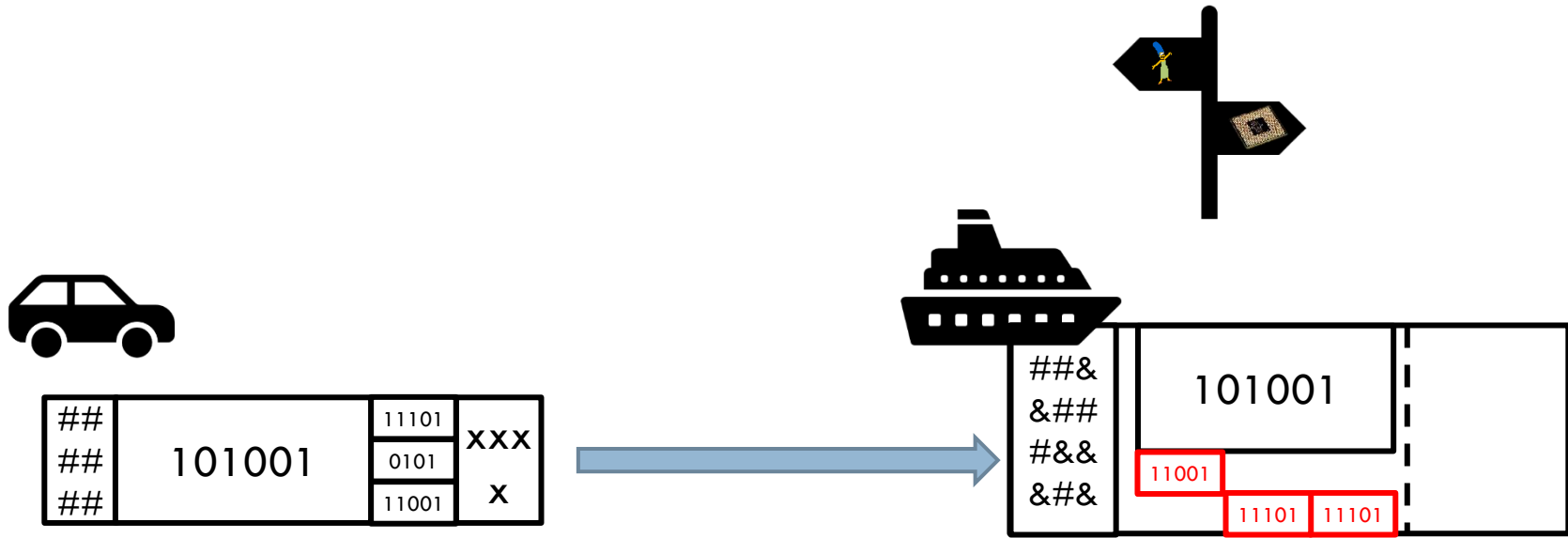
3) Chargement (processus en mémoire vive)



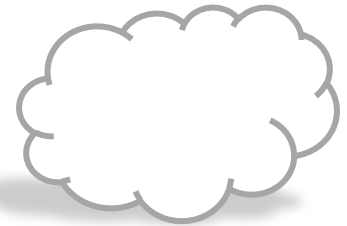
. Processus en RAM : zones de code, zones de données, pile..., 4Gio d'adressage virtuel
 . Notre code objet (transposé ?)



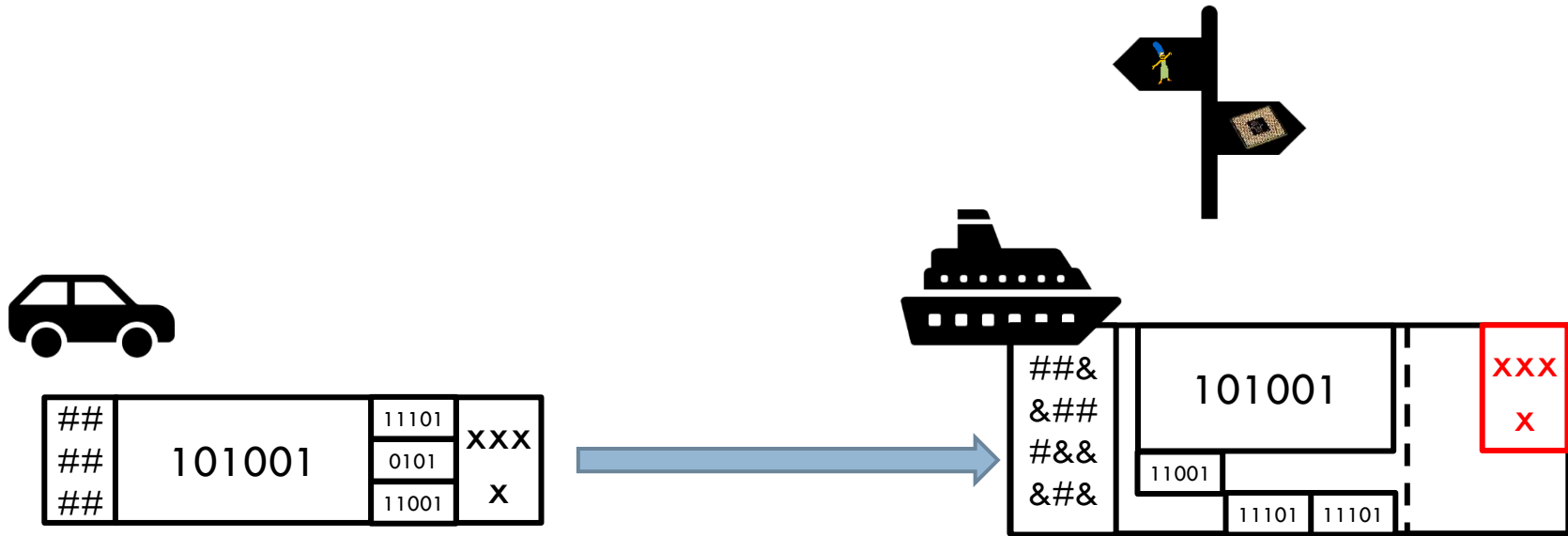
3) Chargement (processus en mémoire vive)



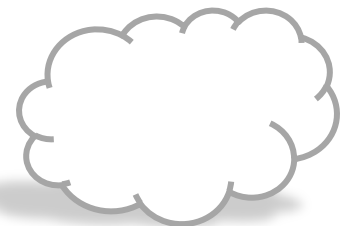
- . Processus en RAM : zones de code, zones de données, pile..., 4Gio d'adressage virtuel
- . Notre code objet
- . Nos bibli. statiques (transposées ?)



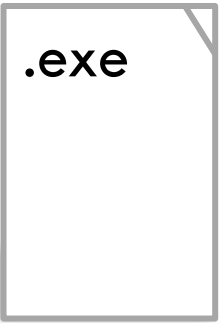
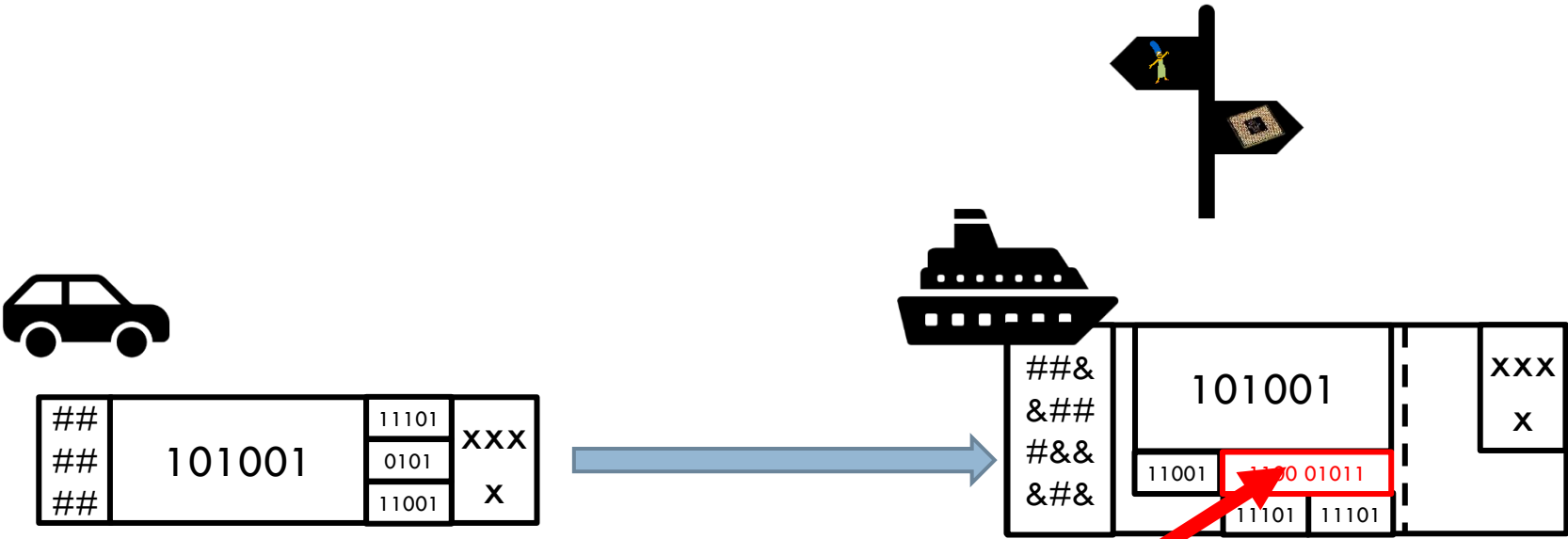
3) Chargement (processus en mémoire vive)



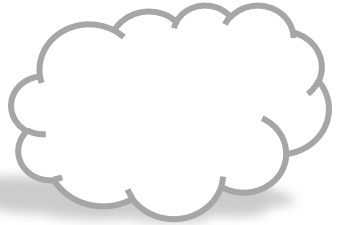
- . Processus en RAM : zones de code, zones de données, pile..., 4Gio d'adressage virtuel
- . Notre code objet
- . Nos bibli. statiques
- . Notre zone de données



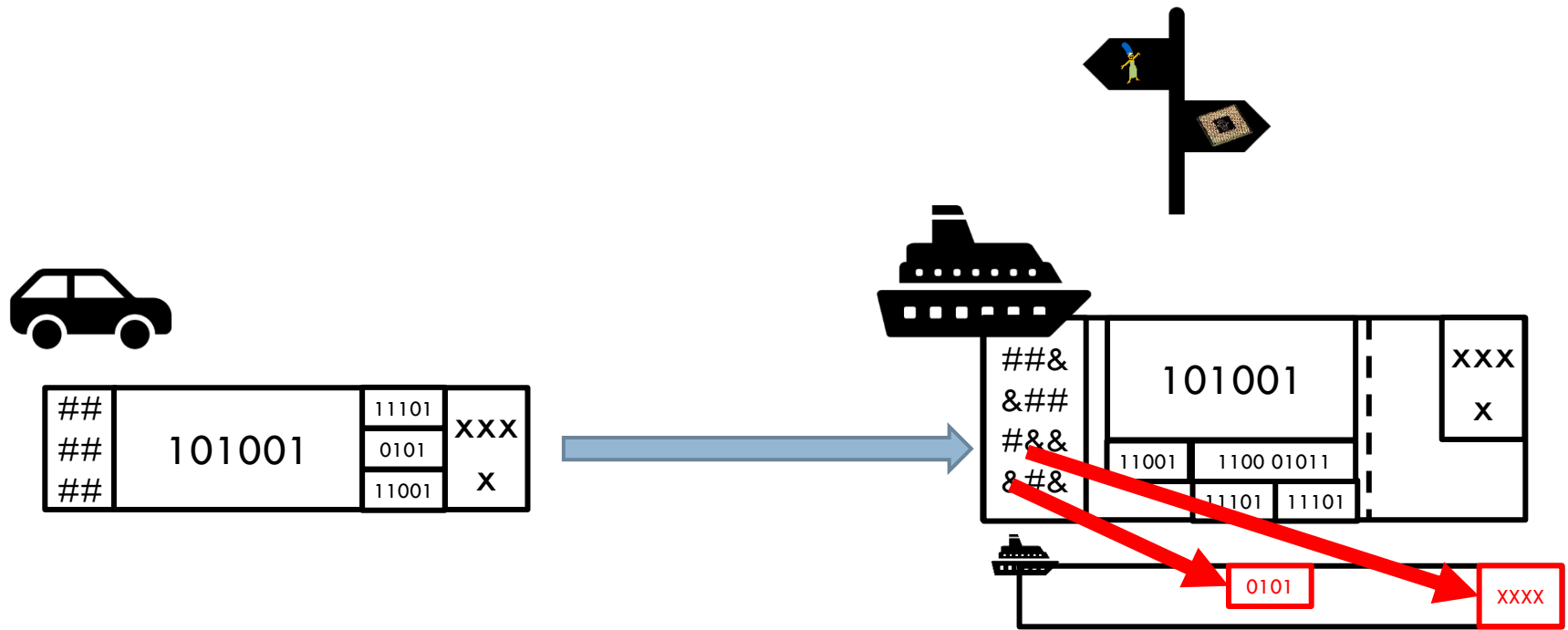
3) Chargement (processus en mémoire vive)



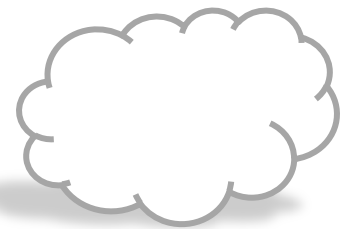
- . Processus en RAM : zones de code, zones de données, pile..., 4Gis d'adressage virtuel
- . Notre code objet
- . Nos bibli. statiques
- . Notre zone de données
- . Bibliothèque(s) dynamique(s)

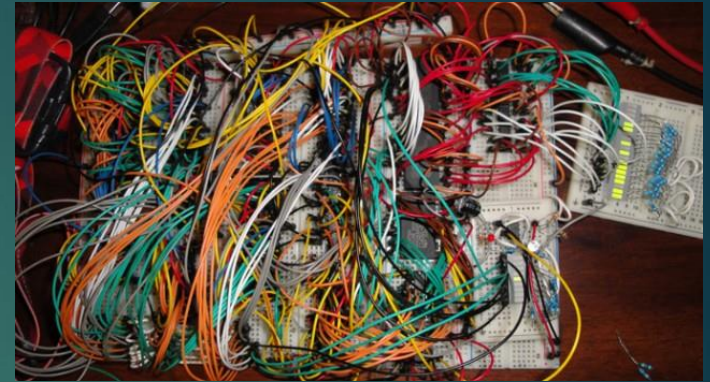
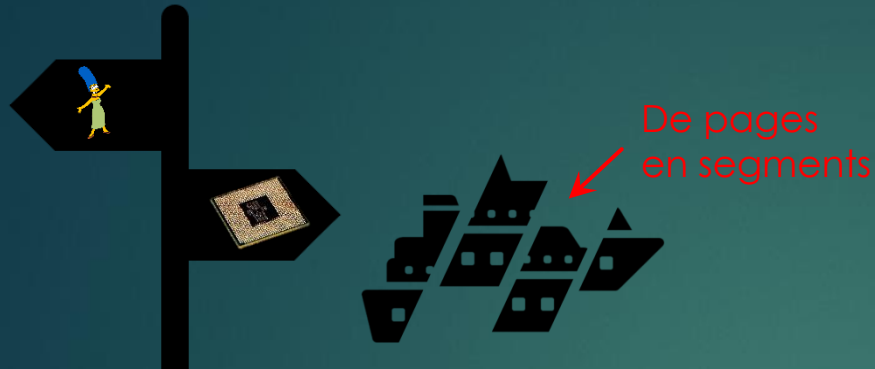


3) Chargement (processus en mémoire vive)



- . Processus en RAM : zones de code, zones de données, pile..., 4Gio d'adressage virtuel
- . Notre code objet
- . Nos bibli. statiques
- . Notre zone de données
- . Bibliothèque(s) dynamique(s)





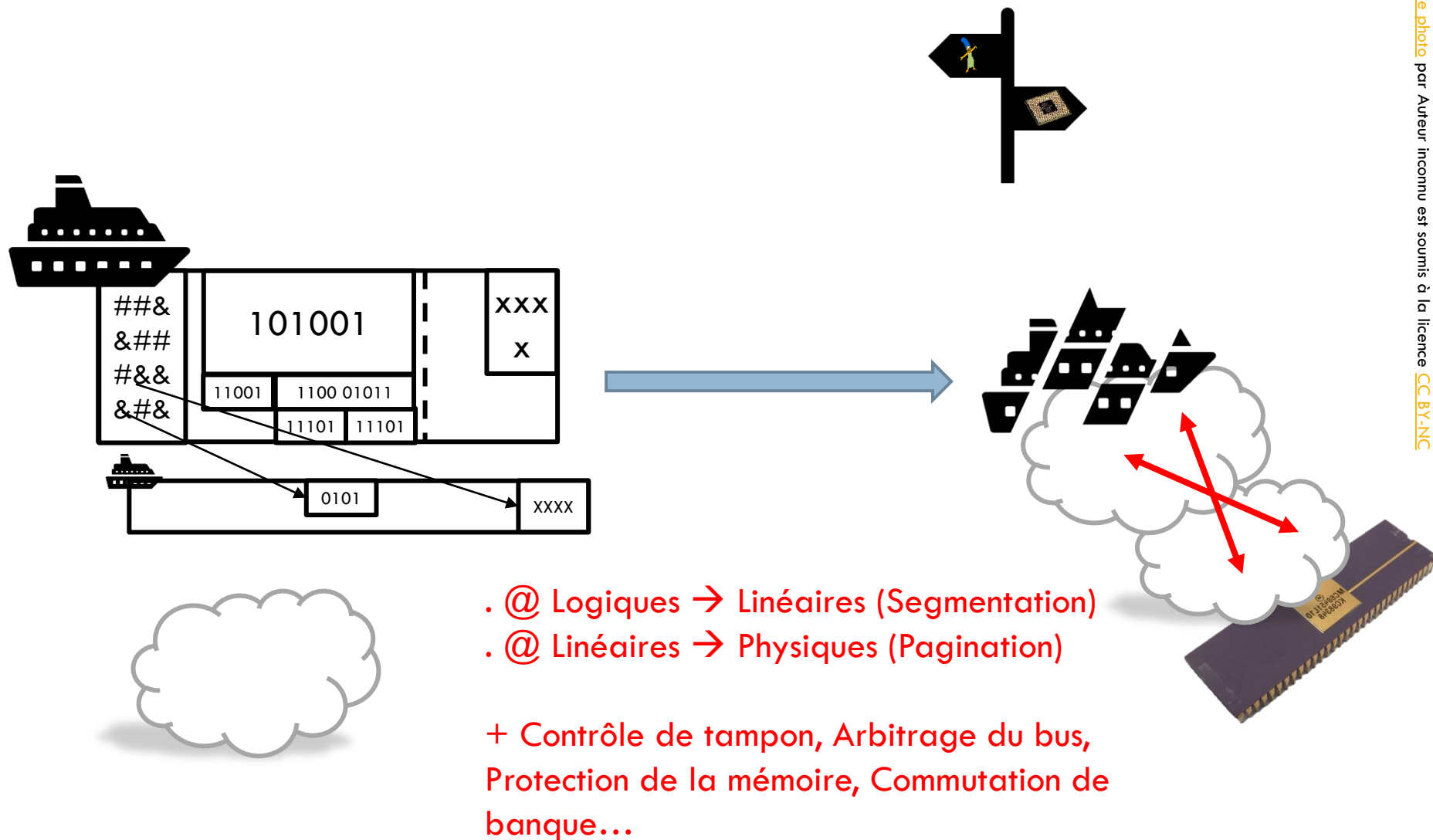
Le fil d'Ariane des adresses relatives...

IV – Mémoire virtuelle, mémoire physique

L'OS ET LE CIRCUIT DÉDIÉ (MMU) ORGANISENT LA MÉMOIRE PHYSIQUE EN PAGES ET/OU SEGMENTS POUR CONTRÔLER LES CONTENUS (INSTRUCTIONS OU DONNÉES) ET LES DROITS D'ACCÈS.

LES ADRESSES MÉMOIRE "VIRTUELLES" DU PROCESSUS ACTIF SONT CONVERTIES À LA VOLÉE EN ADRESSES PHYSIQUE DESTINÉES AU PROCESSEUR.

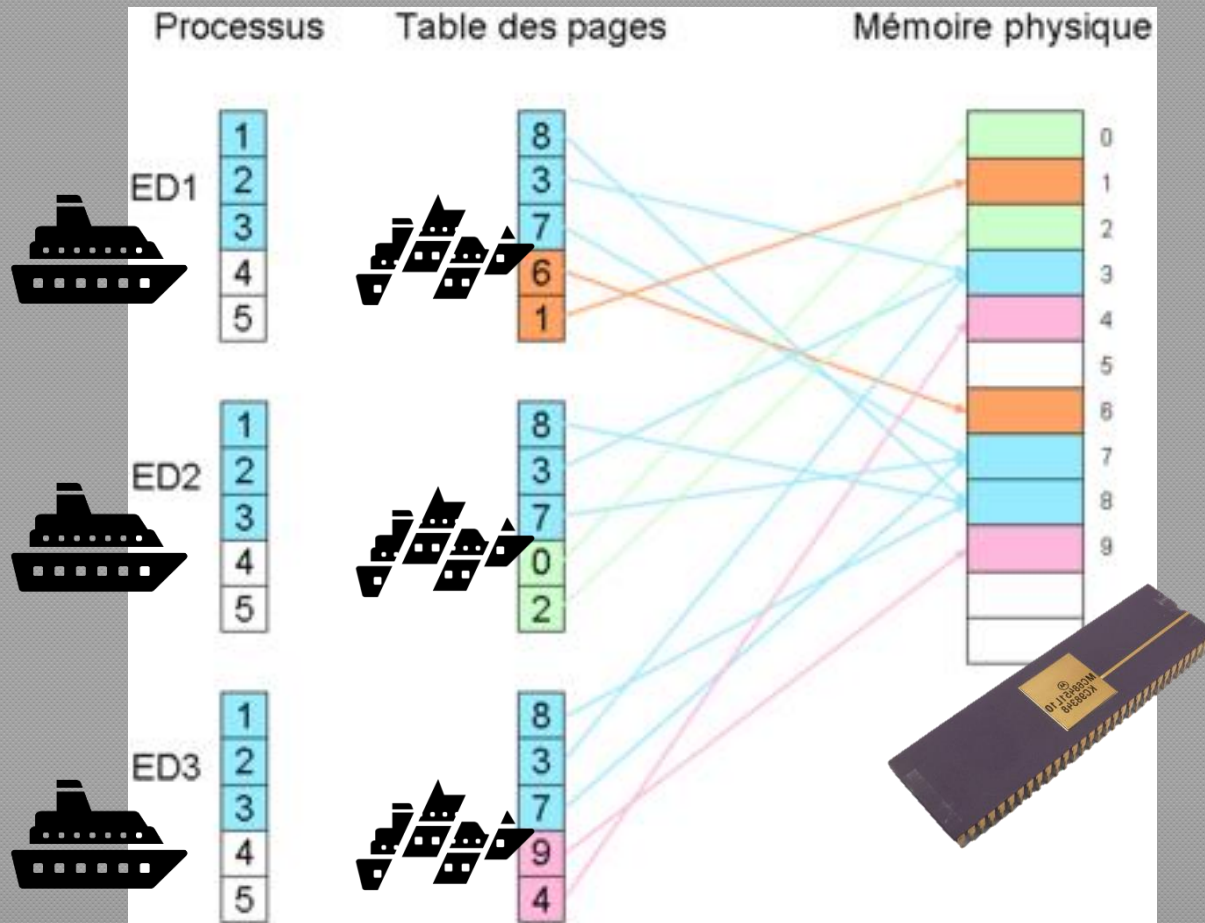
4) Mémoire virtuelle, mémoire physique



4) Mémoire virtuelle, mémoire physique

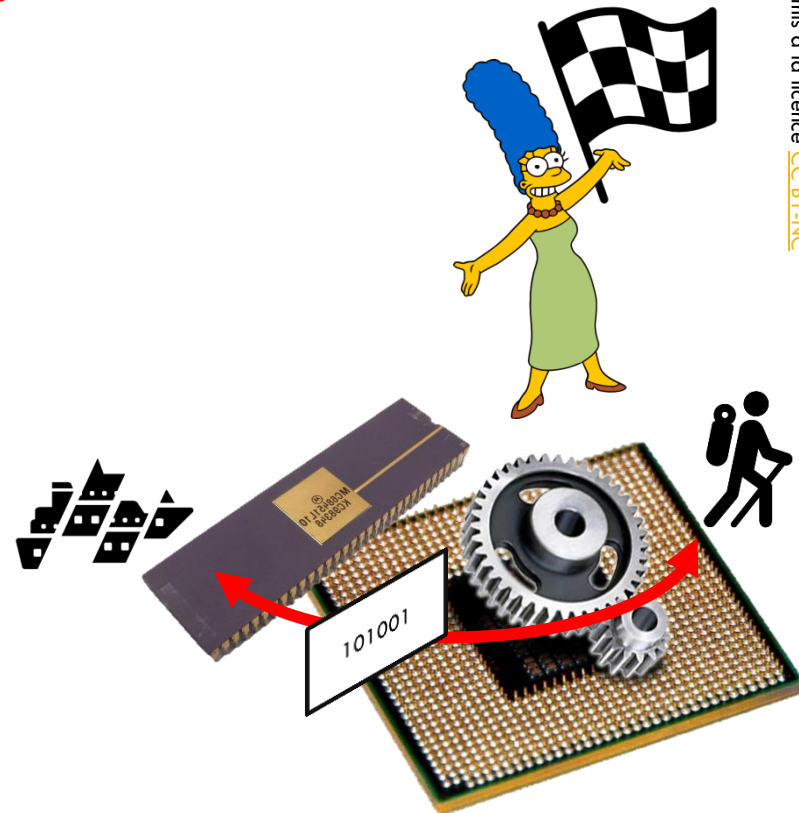
3 processus, constitués de 3 pages de code et 2 pages de données

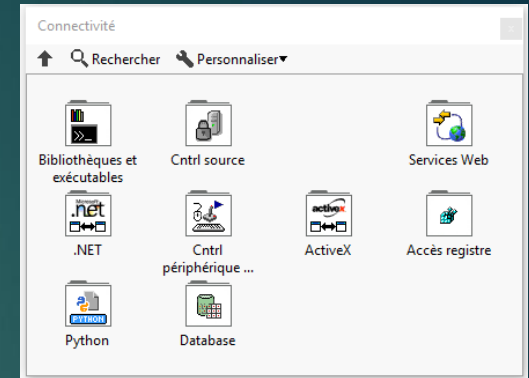
Adresses physiques des pages de code (partagées) et de données (spécifiques).



4) Mémoire virtuelle, mémoire physique

Le processeur va enfin exécuter nos instructions, ça y est, **c'est fini !**

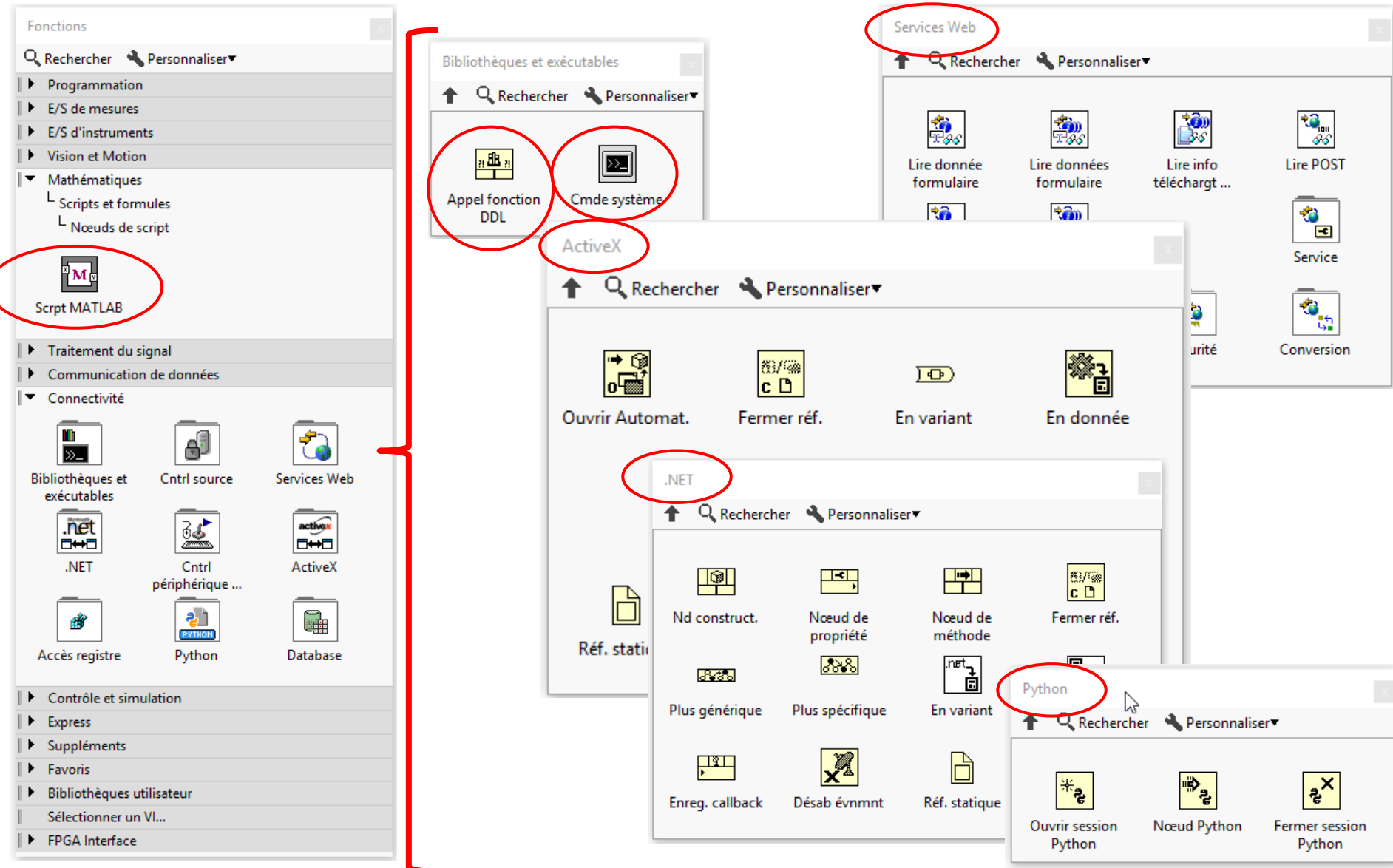




La palette "Connectivité" à l'honneur

V – Quelles interfaces de code à code ?

5) Quelles interfaces de code à code ?



ANNEXES

BIBLIOGRAPHIE

TYPES ET FORMATS DE FICHIERS

ANNEXE I : Bibliographie

"Compilation"

- <https://fr.wikipedia.org/wiki/Compilateur>

"Décoration des noms"

- https://en.wikipedia.org/wiki/Name_mangling

"Edition des liens"

- https://en.wikipedia.org/wiki/Dynamic_linker
- https://fr.wikipedia.org/wiki/%C3%89dition_de_liens

"Code objet"

- https://fr.wikipedia.org/wiki/Fichier_objet
- https://fr.wikipedia.org/wiki/Code_objet
- https://en.wikipedia.org/wiki/Name_mangling
- https://fr.wikipedia.org/wiki/Langage_machine
- <https://fr.wikipedia.org/wiki/Assembleur>

"Bibliothèque logicielle"

- https://fr.wikipedia.org/wiki/Biblioth%C3%A8que_logicielle
- "Static, Shared Dynamic and Loadable Linux Libraries" (yolinux.com)
<http://www.yolinux.com/TUTORIALS/LibraryArchives-StaticAndDynamic.html>
- https://fr.wikipedia.org/wiki/%C3%89dition_de_liens
- [https://en.wikipedia.org/wiki/Library_\(computing\)](https://en.wikipedia.org/wiki/Library_(computing))
- https://fr.wikipedia.org/wiki/Dynamic_Link_Library
- [https://en.wikipedia.org/wiki/Library_\(computing\)](https://en.wikipedia.org/wiki/Library_(computing))

ANNEXE I : Bibliographie

"Exécutable", "Mémoire"

- Les formats exécutables de Windows (Eric Minso)
<http://minso.free.fr/cavinfo/systeme/executable.html>
- https://fr.wikipedia.org/wiki/Fichier_ex%C3%A9cutable
- La mémoire virtuelle (Eric Minso)
<http://minso.free.fr/cavinfo/systeme/memoirevirtuelle.html>
- https://fr.wikipedia.org/wiki/Adressage_m%C3%A9moire
- https://fr.wikipedia.org/wiki/M%C3%A9moire_virtuelle
- <https://fr.wikipedia.org/wiki/X86>
- https://fr.wikipedia.org/wiki/Unit%C3%A9_de_gestion_m%C3%A9moire
- [https://fr.wikipedia.org/wiki/Ex%C3%A9cution_\(informatique\)](https://fr.wikipedia.org/wiki/Ex%C3%A9cution_(informatique))
- https://fr.wikipedia.org/wiki/Portable_Executable

"Interopérabilité des codes"

- https://fr.wikipedia.org/wiki/Application_binary_interface
- https://en.wikipedia.org/wiki/Application_binary_interface
- https://en.wikipedia.org/wiki/Calling_convention
- https://en.wikipedia.org/wiki/Foreign_function_interface
- https://en.wikipedia.org/wiki/Language_binding
- https://en.wikipedia.org/wiki/Component_Object_Model

ANNEXE I : Bibliographie

"Processus"

- [https://fr.wikipedia.org/wiki/Processus_\(informatique\)](https://fr.wikipedia.org/wiki/Processus_(informatique))
- [https://fr.wikipedia.org/wiki/Pile_\(informatique\)](https://fr.wikipedia.org/wiki/Pile_(informatique))
- https://fr.wikipedia.org/wiki/Commutation_de_contexte
- https://fr.wikipedia.org/wiki/Process_control_block

"Interpréteur", "Machine virtuelle"

- <https://fr.wikipedia.org/wiki/Bytecode>
- https://fr.wikipedia.org/wiki/Machine_virtuelle
- https://fr.wikipedia.org/wiki/Langage_de_script
- [https://fr.wikipedia.org/wiki/Interpr%C3%A8te_\(informatique\)](https://fr.wikipedia.org/wiki/Interpr%C3%A8te_(informatique))

ANNEXE II : Types et formats

