

Architecture des ordinateurs





Dimitri Boudier – PRAG ENSICAEN
dimitri.boudier@ensicaen.fr

Avec la participation de :

- Hugo Descoubes (PRAG ENSICAEN)
- Philippe LEFEBVRE (PRAG ENSICAEN)



Except where otherwise noted, this work is licensed under
<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Objectifs

Ce cours s'inscrit dans la continuité de **Systèmes Embarqués**.

En **Systèmes Embarqués** :

Utilisation **poussée** d'une architecture spécifique **simple** (un seul composant, une seule couche logicielle).

→ Observation de l'impact des instructions sur le matériel.

En **Architecture des Ordinateurs** :

Ordinateur de constitution complexe d'un point de vue matériel (plusieurs composants) et logiciel (plusieurs couches)

→ Observations des interactions entre matériel et logiciel.

Objectifs

Et ce cours est complémentaire de celui de **Systèmes d'Exploitation (S8)**.

En **Architecture des Ordinateurs** :

Étude d'un ordinateur en user-space :

- comment sont construites les applications créées par un développeur ?
- comment ces applications s'exécutent-elles ?
- quelle est la marge de manœuvre du développeur sur l'exécution de ses applications ?

En **Systèmes d'Exploitation** :

Étude d'un ordinateur en kernel-space :

- comment le système d'exploitation organise-t-il l'exécution de plusieurs applications en simultané ?
- comment le système d'exploitation gère les cas particuliers (division par zéro, ...)?

Objectifs

À la fin du module, vous devrez :



Maîtriser toutes les étapes de la vie d'un programme

De la rédaction (en C) à son exécution, en passant par la compilation



Comprendre les interactions entre le matériel et le logiciel

Fichiers stockés sur la mémoire de masse

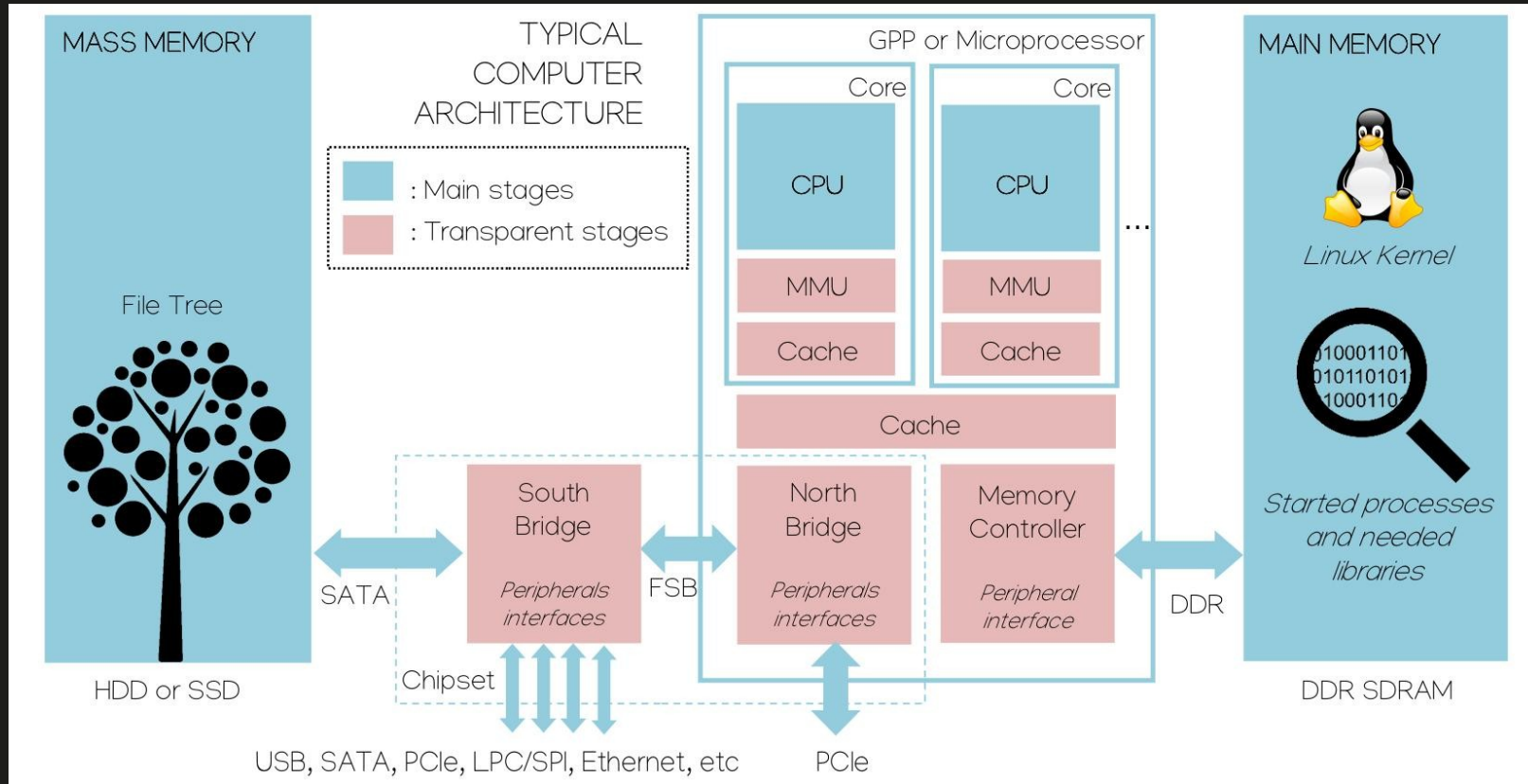
Programmes exécutés par le processeur central

Données et instructions stockées en mémoire de travail

L'idée est de vous faire porter un regard critique sur le code C que vous rédigez, en comprenant l'impact de vos lignes de code sur la machine.



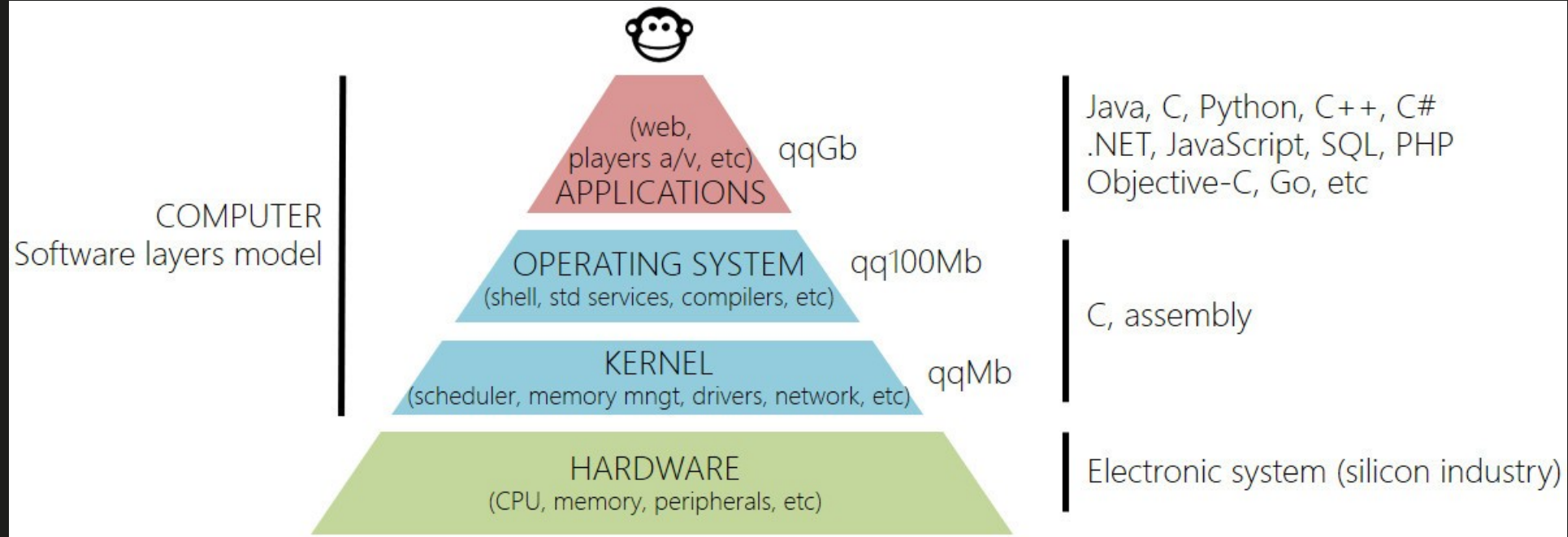
Dans ce cours, nous parlerons évidemment de l'architecture matérielle d'un ordinateur.





Mais pour cette année, nous ne parlerons que brièvement de son architecture logicielle.

Les systèmes d'exploitations sont vus en 2A et 3A SATE.



Notre principal objectif sera de comprendre l'interaction entre le logiciel et le matériel.

Compilation

Pourquoi compile-t-on un programme ?

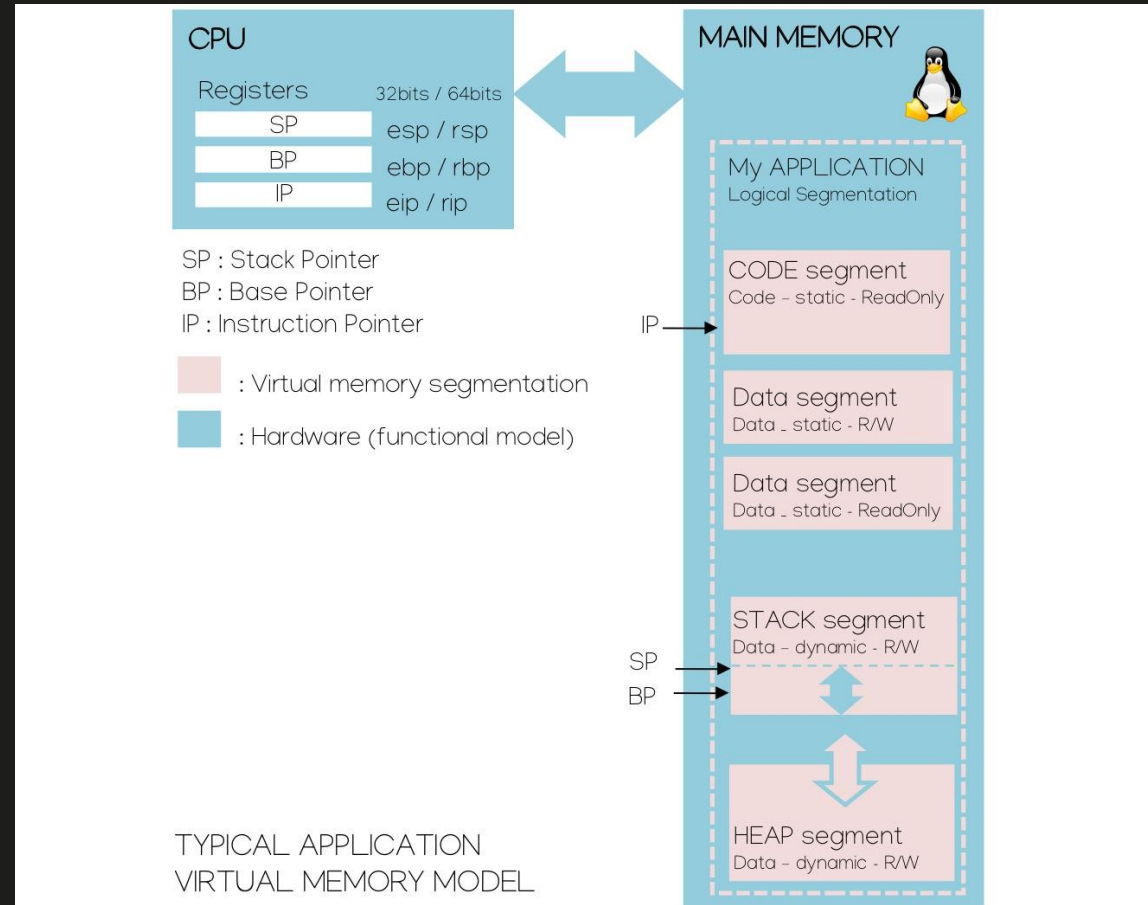
Que se passe-t-il pendant la compilation ?

Exécution

Comment s'exécute un programme ?

Comment la RAM est-elle utilisée ?

Comment le GPP accède-t-il aux données et instructions d'un programme ?



Ressources

Toutes les ressources se trouvent sur Moodle, en accès libre :



<https://foad.ensicaen.fr/course/view.php?id=696>

- Supports de CM, sujets de TP
- Consignes pour les travaux préparatoires
- Conseils et outils pour travailler à distance
- Guides pour l'installation d'une machine virtuelle et Ubuntu, ...

Ressources

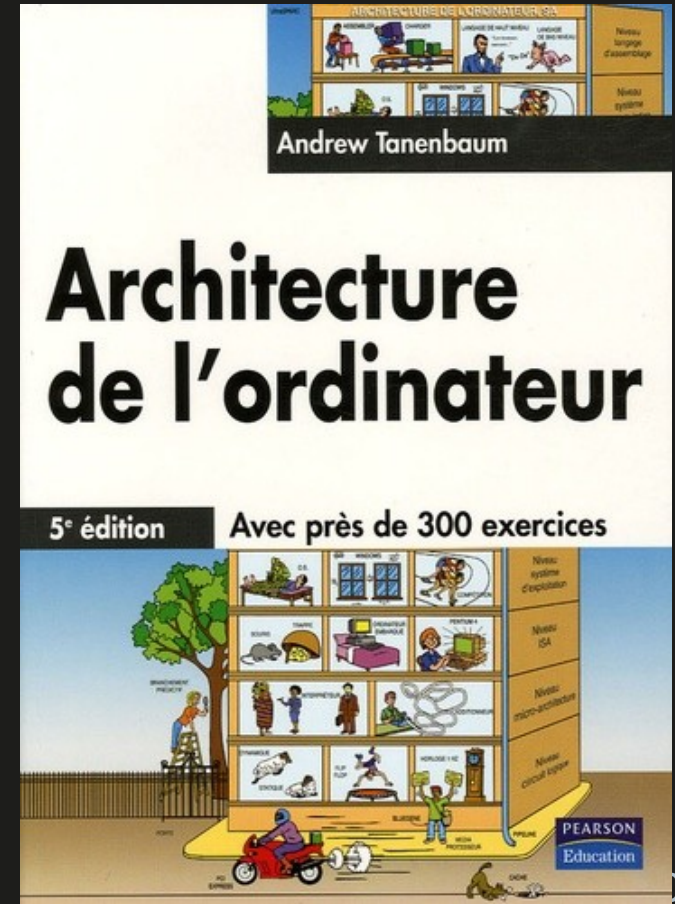
Le livre d'Andrew Tanenbaum est une pépite.
« Structured Computer Organization », 5th or 6th dition.

Andrew Tanenbaum est le créateur de l'OS MINIX.

Il a aussi fait d'autres livres sur les OS et sur les réseaux.

Dans ce livre :

- Des rappels de 1^{ère} année :
 - Circuits logiques et numération, vus en Électronique Numérique
 - CPU, vu en Systèmes Embarqués
- Des compléments et exemples en lien avec ce cours
 - Et avec le cours « Systèmes d'Exploitation » vu avec P. Lefebvre
- Des concepts qui seront abordés en 3A



6 x 1h CM → Dimitri Boudier

7 x 2h TP → Dimitri Boudier & Philippe Lefebvre

Un examen sur table d'1h30

Portera sur ce qui a été vu en cours et en TP (GPP, compilation, mémoire, stack, ...)

Polycopiés interdits

Une feuille manuscrite A4 recto-verso autorisée