


PRELUDE

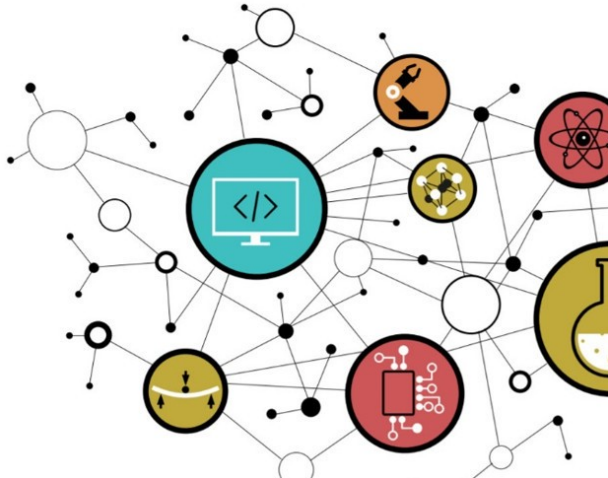
hugo descoubes – enseignant Systèmes Embarqués – ENSICAEN - France
Outils - GNU\Linux Ubuntu 20.04 LTS – LibreOffice 6.4.6.2 – 2022

Électronicien spécialisé en Systèmes Embarqués, je suis à votre écoute et ferais de mon mieux pour vous accompagner et vous aiguiller dans votre projet professionnel !

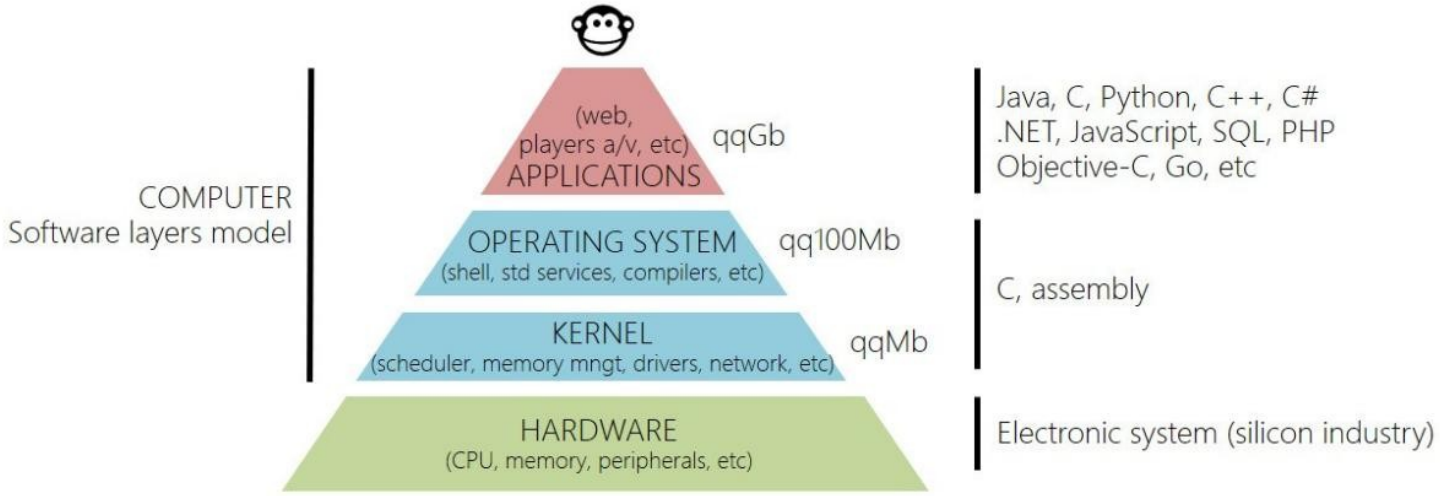
- Bureau en A202 (accès digicode par la salle A203 - *carré + A203 + triangle*) – 02 31 45 27 61

- hugo.descoubes@ensicaen.fr

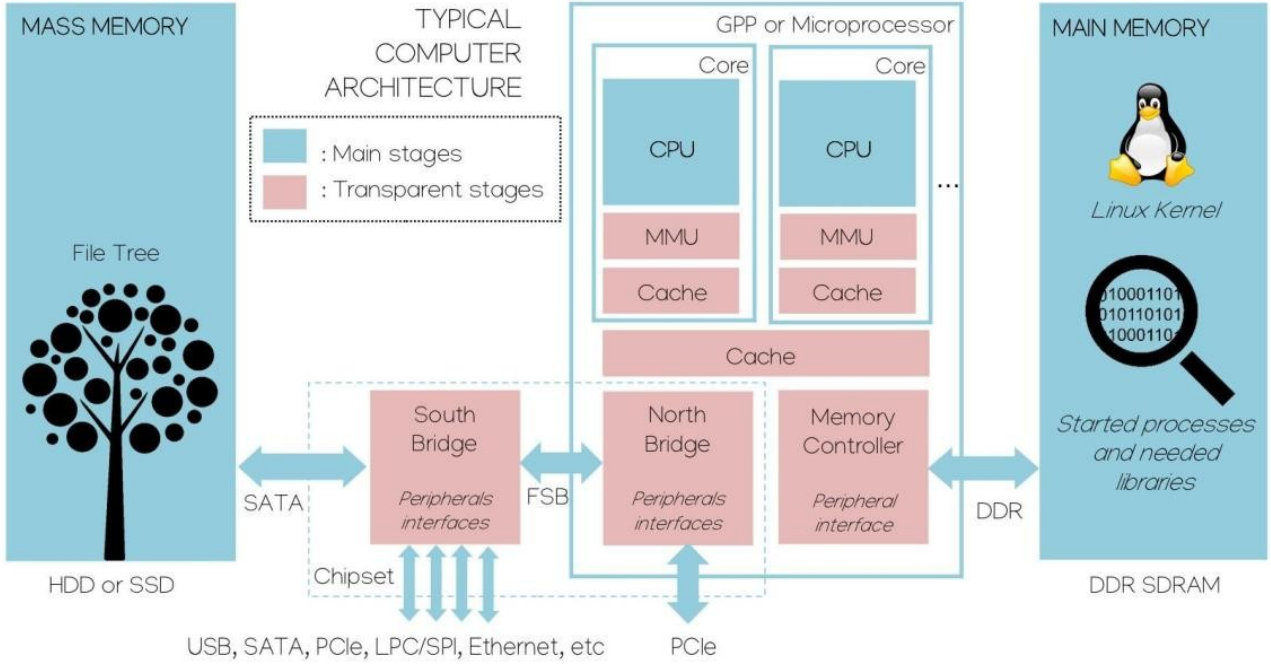
- Objectifs
- Ressources pédagogiques
- Évaluations des compétences
- Pré-requis



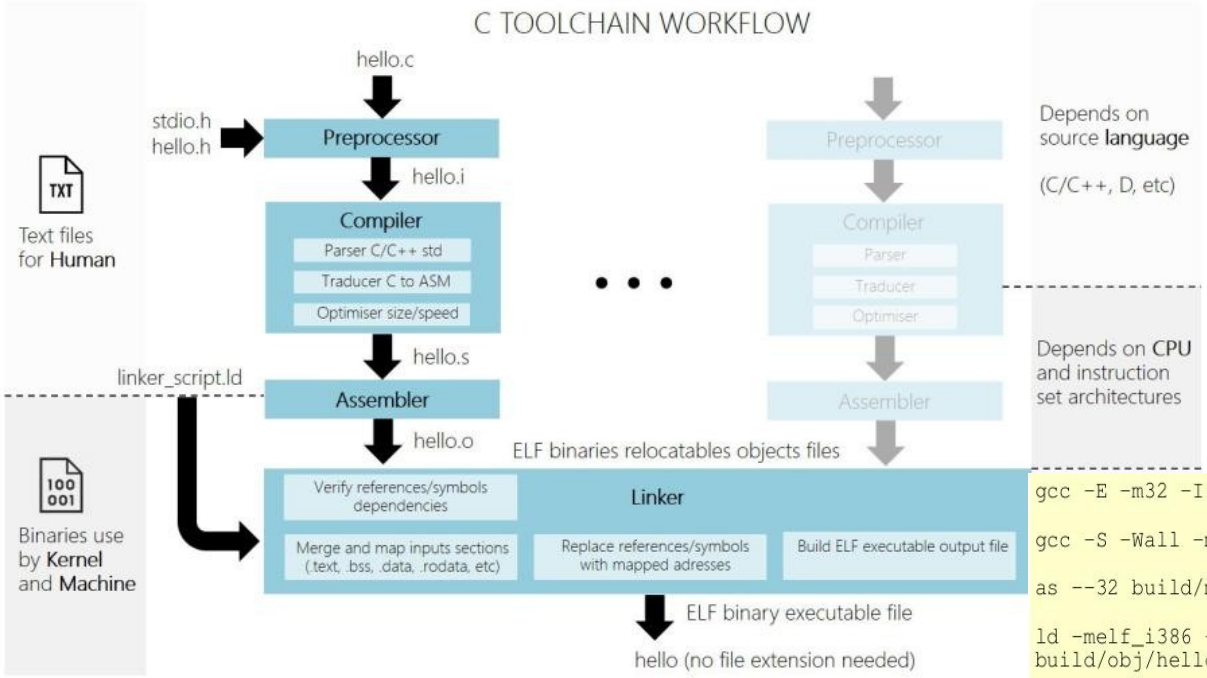
Une bonne maîtrise des langages de programmation système (C et ASM), du processus de compilation et d'édition des liens, du travail de cloisonnement du noyau système au chargement d'un programme en mémoire principale (segmentation et pagination), ainsi que la compréhension des rôles des principaux composants matériels et du fonctionnement global de l'ordinateur sont des bases fondamentales pour un développeur logiciel, même haut niveau (JAVA, C++, etc).



- Connaître les principaux composants matériels d'un ordinateur. Comprendre l'architecture et le fonctionnement global d'un système numérique d'information géré par un système d'exploitation évolué exploitant une MMU et un espace mémoire virtualisé (Linux, NT, etc)



- Comprendre le processus de compilation et d'édition des liens. Connaître les principaux outils d'analyse et de compilation du projet GNU (gcc, as, ld, objdump, readelf, strip, ar, etc)

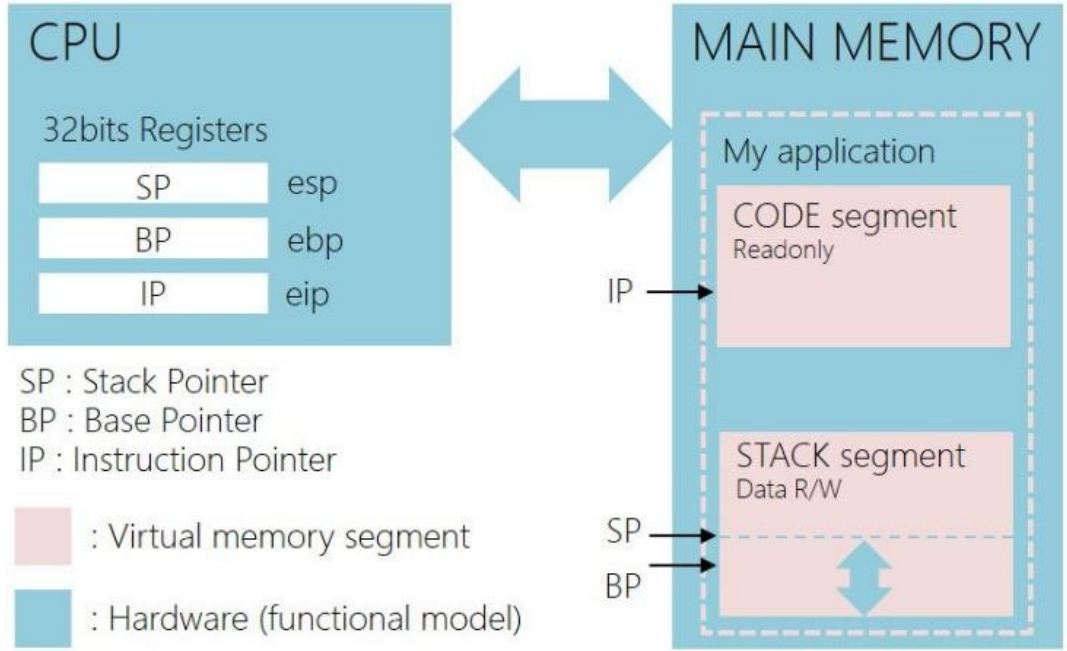



```
gcc -E -m32 -I./inc src/hello.c > build/misc/hello.i

gcc -S -Wall -m32 build/misc/hello.i -o build/misc/hello.s

as --32 build/misc/hello.s -o build/obj/hello.o

ld -melf_i386 -T build/script/linker_script.ld build/obj/crt0.o
build/obj/hello.o -o build/bin/hello
```

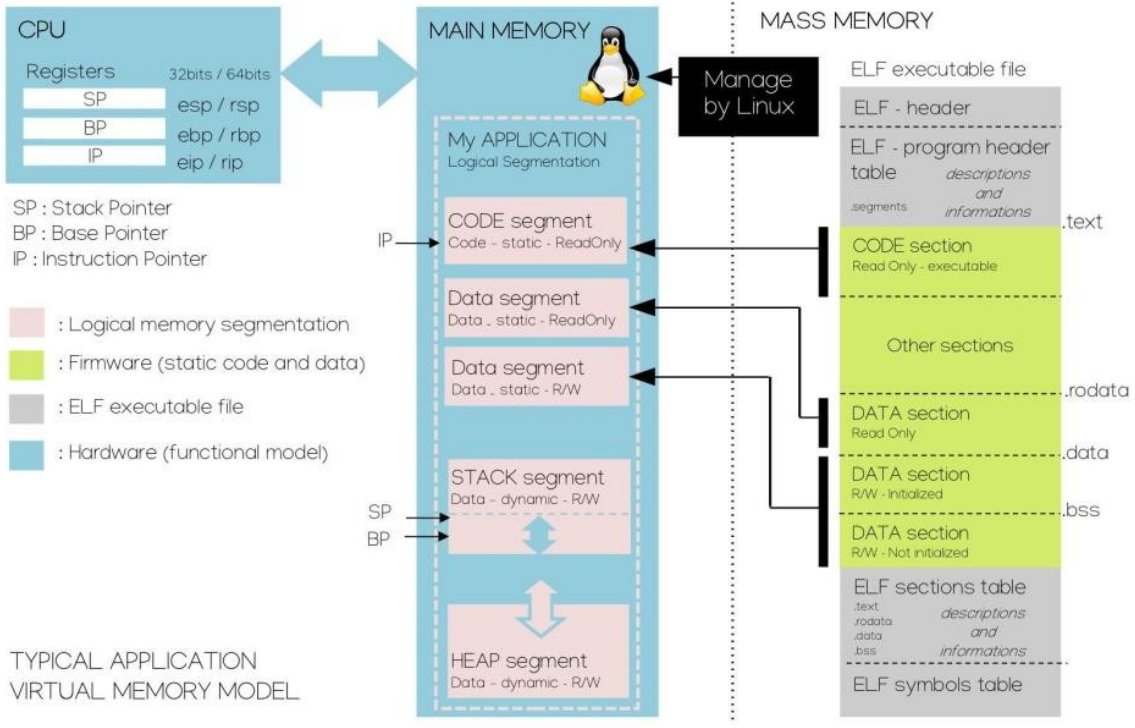
- Analyser un programme ASM x86_64. Comprendre le mécanisme de gestion des variables locales sur la pile réalisé par les outils de compilation, par le système ainsi que la machine

```
int main (void)
{

return 0 ;
}
```

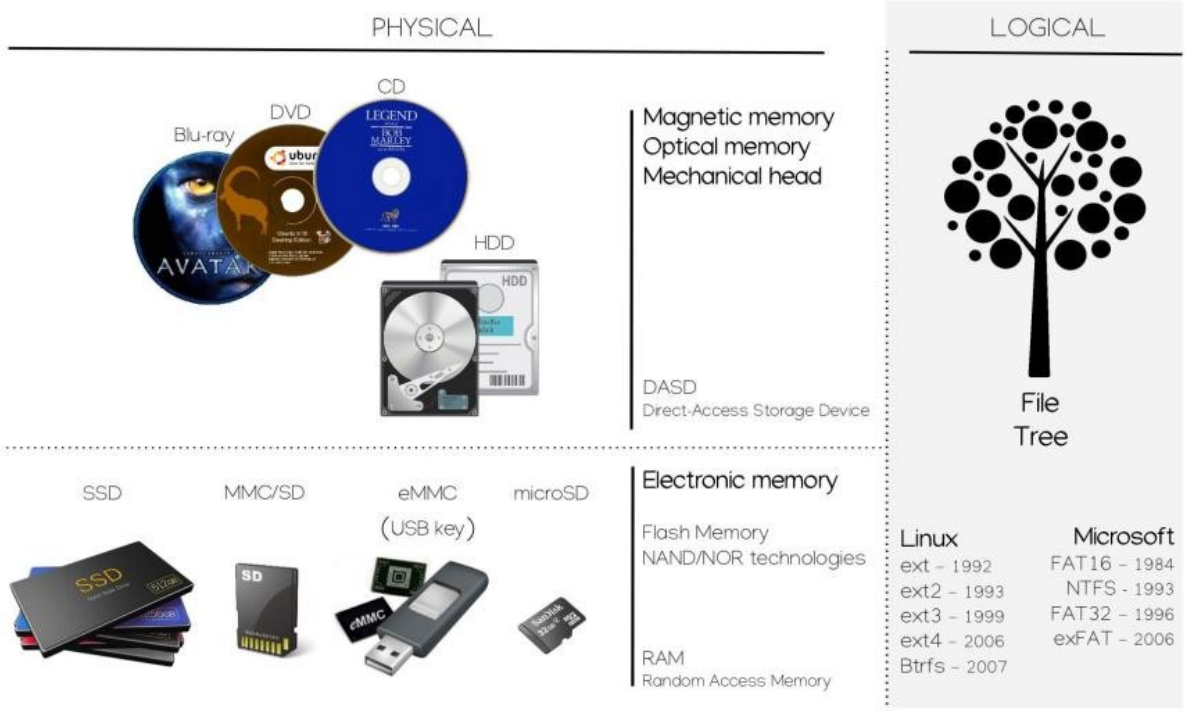
```
main:
        pushl   %ebp
        movl    %esp,%ebp
        movl    $0,%eax
        popl    %ebp
        ret
```



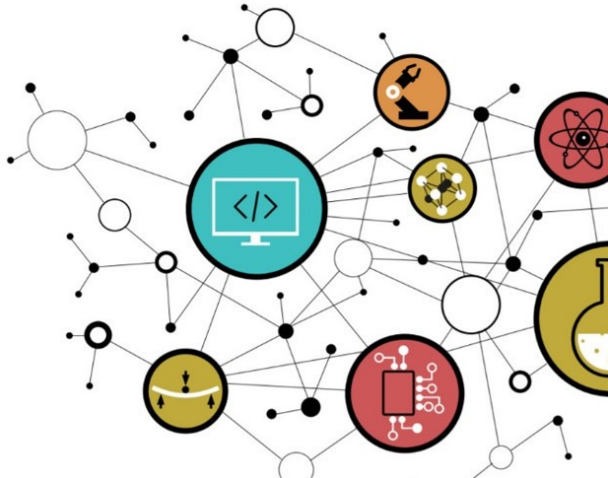
- Comprendre le processus et l'environnement d'exécution d'une application logicielle. Connaître les limitations mémoire imposées par le système et l'ordinateur (segmentation et pagination)



- Comprendre l'architecture logique des supports de stockage de masse (table des partitions, systèmes de fichiers, etc). Être apte à préparer un média de masse pour un besoin spécifique.



- Objectifs
- **Ressources pédagogiques**
- Évaluations des compétences
- Pré-requis



- Archive complète de travail Cours/TP sur la plateforme moodle ENSICAEN : **arch.zip**

https://foad.ensicaen.fr/course/view.php?id=99

- Polycopiés séparés de cours et de TP

- Les TP peuvent être réalisés dans les salles informatique du bâtiment E ou en salles A203 et A201 (digicode carré + A203 ou A201 + triangle)

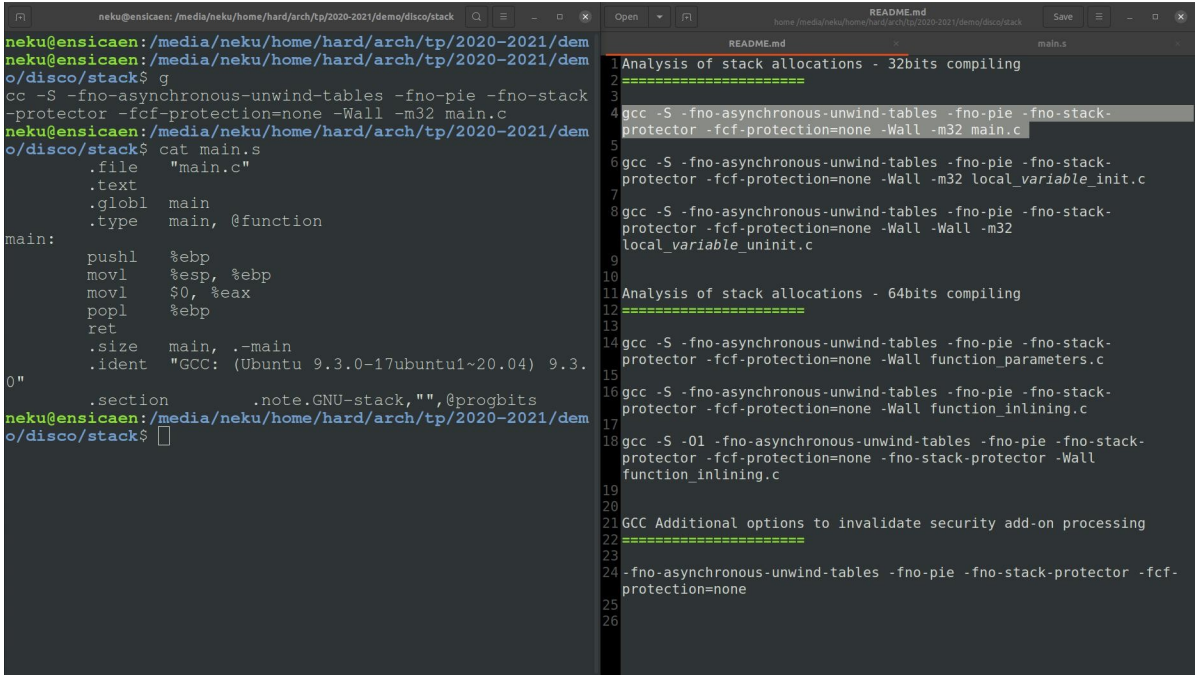
- Nous vous conseillons néanmoins d'utiliser vos machines personnelles avec un système GNU\Linux 64bits. Afin d'utiliser les même configurations système que l'école, nous vous conseillons d'installer un système Ubuntu 20.04 LTS. Vous trouverez ci-dessous un tutoriel ENSICAEN pour vous aiguiller dans vos installations

https://ubuntu.com/download/desktop?version=20.04&architecture=amd64

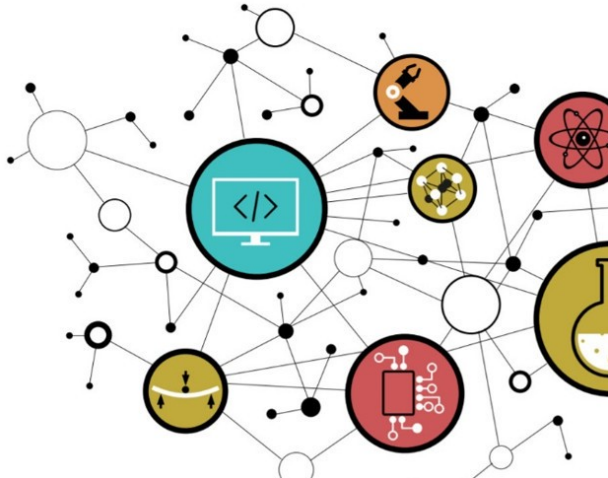
https://foad.ensicaen.fr/mod/page/view.php?id=22125

https://foad.ensicaen.fr/mod/page/view.php?id=25095

- L'environnement de travail se voudra volontairement minimaliste. Un système GNU\Linux 64bits, quelques programmes sources C et ASM x86_64 à analyser, le shell, une toolchain GNU GCC, etc et nous pourrons explorer les entrailles de notre ordinateur ...



- Objectifs
- Ressources pédagogiques
- **Évaluations des compétences**
- Pré-requis



- Partiel 30mn QCM sur moodle :
  - Compilation et édition des liens
  - Assembleur 32bits x86 et 64bits x86_64

- Examen 1h30 à l'écrit sur table :
  - 7pts : 3-4 questions ouvertes d'ordre général. Illustrer à l'aide d'un schéma exhaustif avec définitions et fonctionnement du système à présenter. Peut-être vu comme un échange d'ingénieur à ingénieur.
  - 13pts : Exercice de traduction et de retro-ingénierie ASM x86_64 vers langage C. Analyse d'un programme ASM x86_64 et du fichier objet associé au format ELF. Analyse du comportement et trace du contenu de la pile

Merci !