

CM1:

allouer la ressource Centrale (UC)

● CPU peut être dispatcher entre plusieurs user
un processus allouer à un processeur pour un quantum de temps
(illusion //)

OS doit assurer isolement (ex: mem centrale partager ≠ proc
⇒ changement de

Entre processus attend hop longs (formine) ^{de processeur} ordonnanceur par
le scheduler

● C (man 3) tamponne entrée/sortie ⇒ 3 printf

(man 2) 3 printf → 3 write
(+ long)

→ 1 write
(appel système)

shell: permet donner langage de manipulation du système via
interpréteur commande.

Conso: \$ ls
↑ fait au niveau shell

● application: prog externe au shell (^{Fichier} ~~en~~ binaire exécutable)
qui peut être exécuter grâce au shell qui va les chercher via
var env: PATH (liste des répertoires contenant appli)

interpréter (langage): traduction se fait en temps réel via interpréter
(il analyse, traduit et exécut prog). Diffère de l'exéc dynamique
du prog par un prog (interprète) plutôt que par la conversion en un
autre langage (^{en} langage machine) entre séparément du temps conversion
et du temps d'exéc qui sont simultanés.

cycle d'un interprète:

- lire analyser une instruction (ou exp)
- si ~~exp~~ instruction syntaxiquement correcte, l'exécuter (ou si exp l'évalue)
- passer à l'instruction suivante.

Un pas un (step) \Rightarrow plus lent, bien que la plus part interprète exécute un arbre syntaxique (sorte de représentation interne des données).

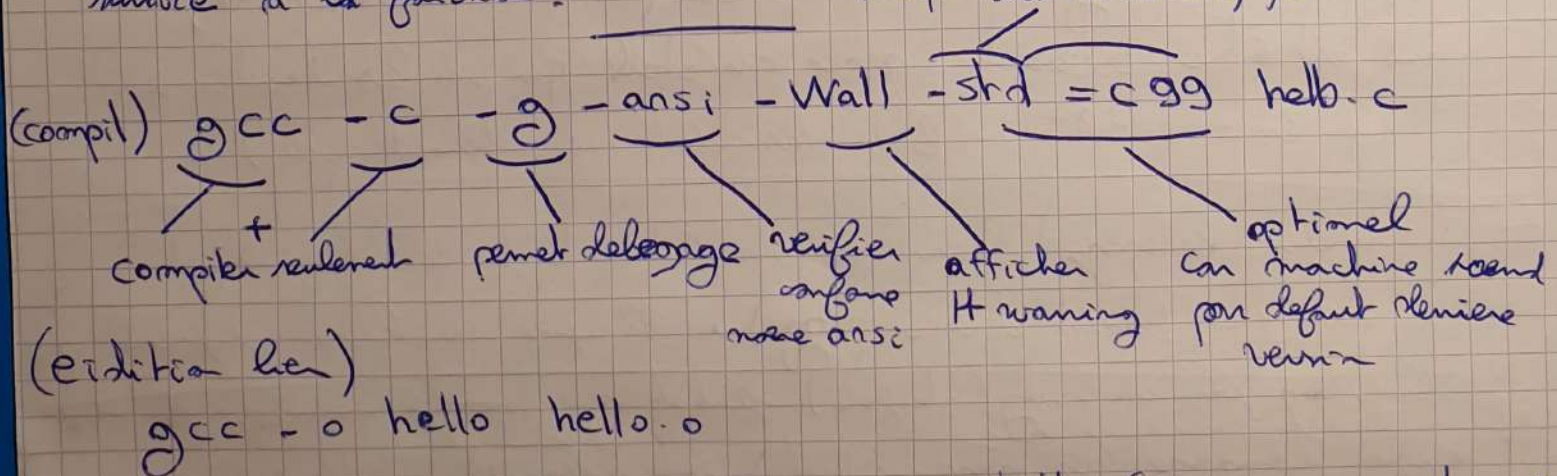
interprète utilise globalement un outil compil (syntaxe abstraite, lexical analy, ...)

interprète avantage: plus portable (pas besoin recompiler donc moins de temps perdu, plus tolérant ...) il suffit interprète sur \neq plateforme

processus: instance d'un programme qui est exécuté
"programme peut être vue comme la partie statique ou le squelette"
plusieurs processus peuvent ^{à la fois} travailler sur un prog donc prog pas id suffisant \Rightarrow PID (processus ID).

header inclu dans le source les déclarations \neq body (dans lib std)

var static \Rightarrow on regret donner var global mais local scope
rattaché à la fonction.
indique version du langage C



éditeur lien hello.o pour fabriquer binaire exéc hello (tjs sans extension)
sur windows il se termine par .exe).

on utilise gcc pour éditer lien car ~~cel~~ est un appel assez compliqué
(et donc pour pas se planter on laisse gcc le faire).
+opti

Édition lien

"mm" permet de connaître les symboles utilisés et définis par un binaire:

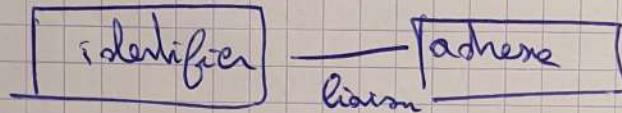
"d": donnée \Rightarrow "data"; "T": texte \Rightarrow "text"

"U": undefined.

~~0000 lol~~ d ex:

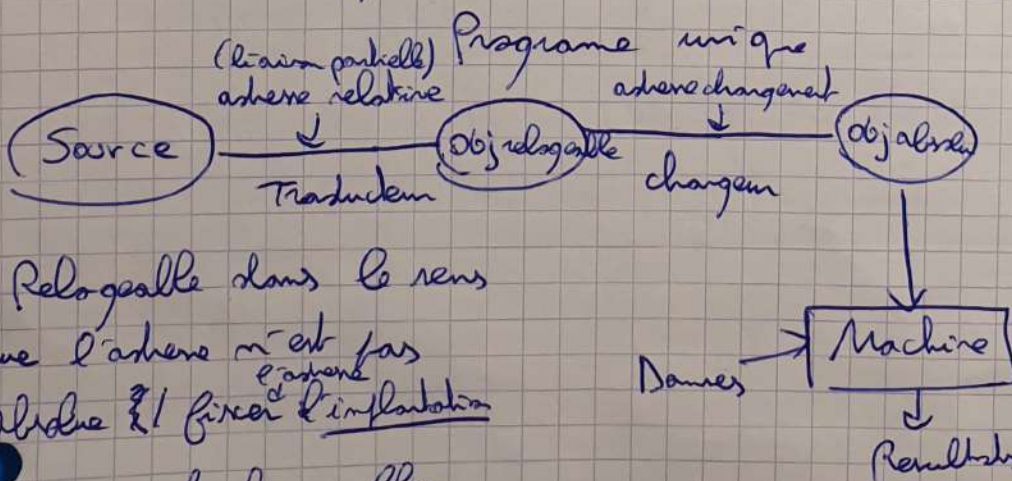
0000 d lol
0000 T dire-str
000 U printf
T main

On désigne toute op qui permet de passer du nom d'un objet informatique à sa représentation physique



Édition lien: phase ayant pour but d'établir la liaison des ref extérieures (ref obj bibliothèque ou obj définies dans modules compilés séparément)

Chargement: remplace les adresses relatives à des début de bloc logique par adresses absolues. en faisant l'adresse



Relogable dans le sens que l'adresse n'est pas absolue & on fixe l'implantation en mem de chaque bloc logique

< => header système ; " => header personnel

dans /usr/bin/include/;

dans . (rep courant)

si rep particulier chemin absolu / relatif

static durée de vie = durée vie processus.

et privée au fichier (utile pour le fichier mais pas à l'app)

variable local a m durée de vie scope (fonction car installer pile exec quand on appelle une f on "installe" un bloc de pile (une frame) qui correspond à la fonction. Une fois la fonction finit on fait passer le ptr de lecture de pile après le bloc (mais ^{renvoie} variable tjrs là : variable globale car non dépile m si plus pointer)

processus ^{segment} à mémoire allouée :

- pile exec (pointe vers fort)
- tas (^{durée} dynam) - segment de code (fichier binaire exec)
- segment mémoire statique

41.00

Quand un processus s'exéc il utilise les registres du processeur
Parmi lesquels :

- Contexte d'exécution /
- le compteur ordinal : il pointe sur l'instruction en cours d'exéc (dans segment code).
 - ptr de pile : pointe vers sommet pile exec (bloc courant ansab - Fonction courante qui est entrain de s'exécuter).
 - ptrs segments : qui pointe vers le segment mem (voir précédent)
 - ...

Quand processus se termine (quantum de temps du) on sauvegarde la valeur des registres (du processeur) dans le contexte d'exéc (qui

une partie de la mem centrale) puis un autre proc installe son contexte dans les registres proc ---

Préemption processus : déterminée par le scheduler de l'os