- un processus
    - représente l'exécution d'un programme séquentiel dynamiquement , un microprocesseur mono cœur ne peut exécuter qu'un processus à la fois.
    - La définition exacte d'un processus, c'est « *programme en cours d'exécution auquel est associé un environnement processeur et un environnement mémoire*. » En effet, au cours de son exécution, les instructions d'un programme modifient les valeurs des registres (le compteur ordinal, le registre d'état...) ainsi que le contenu de la pile.
- Le parallélisme réel (le cas d'un system multi processeur)
    - Deux processus qui fonctionnent réellement en parallèle , un processus occupe le microprocesseur (1) et le deuxième occupe le microprocesseur (2)
- Le pseudo parallélisme :
    - Chaque processeur occupe un par un le microprocesseur pendant un quantum de temps tellement petit qu'on a l'impression d'un parallélisme réel
- La communication :
    - La communication entre 2 processus pour transférer des informations, sois à travers un canal ou une variable
- La synchronisation :
    - La coordination entre les processus pour l'utilisation des ressource d'une manière contrôlée
- L'inter-blocage
    - C'est un des raison qui bloque le programme, c'est le cas où on a deux processus , chacun d'eux est en train d'utiliser une ressource , et quand les deux terminent leurs utilisations, il ne libre par la ressource , et attendent que la ressource de l'autre soit libérer
- Gaspillage du temps de microprocesseur :
    - si un processus est bloqué (il attend qu'une ressource soit libre) , et le processeur donne a ce processus un quantum entier , c'est un gaspillage de temps du processeur
- Famine :
    - un processus en famine est un processus qui n'a pas eu son tour pour être executé par le processeur
- Un sémaphore
    - une structure composée d'une valeur entière positive et d'une file d'attente (FIFO) , et c'est un outil de synchronisation
- Un thread (tache en français)
    - un fil d'instructions à l'intérieur d'un processus , appelé aussi processus léger ou activité
    - La communication entre les threads est plus rapide qu'entre les processus, puisqu'ils partagent le même espace alors que les processus ont chacun son espace mémoire personnel.
- Différence entre thread et processus :
    - Il faut savoir que le principal avantage des threads par rapport aux processus, c'est la facilité et la rapidité de leur création. En effet, tous les threads d'un même processus partagent le même espace d'adressage, et donc toutes les variables. Cela évite donc l'allocation de tous ces espaces lors de la création, et il est à noter

que, sur de nombreux systèmes, la création d'un thread est environ cent fois plus rapide que celle d'un processus.

Au-delà de la création, la superposition de l'exécution des activités dans une même application permet une importante accélération quant au fonctionnement de cette dernière.

- La différence entre processus et programme :
    - Un programme, c'est une suite d'instructions (notion **statique**), tandis qu'un processus, c'est l'image du contenu des registres et de la mémoire centrale (notion **dynamique**).
- les types du multitâche :
    - multitâche préemptif : le système d'exploitation garde le contrôle et se réserve le droit de fermer l'application.
    - multitâche coopératif : le système d'exploitation laisse les applications gérer.

- les états d'un processus :
    - exécution (R pour running) : le processus est en cours d'exécution ;
    - sommeil (S pour sleeping) : dans un multitâche coopératif, quand il rend la main ; ou dans un multitâche préemptif, quand il est interrompu au bout d'un quantum de temps ;
    - blocké : attend qu' une ressource soit disponible
    - arrêt (T pour Terminated) : le processus a été temporairement arrêté par un signal. Il ne s'exécute plus et ne réagira qu'à un signal de redémarrage ;
    - zombie (Z pour … zombie) : le processus s'est terminé, mais son père n'a pas encore lu son code de retour.
- PATH : Lorsqu'on lance une commande dans la console, le système va chercher l'exécutable dans les chemins donnés dans le PATH
- section critique : est une portion de code dans laquelle il doit être garanti qu'il n'y aura jamais plus d'un thread simultanément. Il est nécessaire d'utiliser des sections critiques lorsqu'il y a accès à des ressources partagées par plusieurs threads.

source :

- https://openclassrooms.com/courses/la-programmation-systeme-en-c-sous-unix/