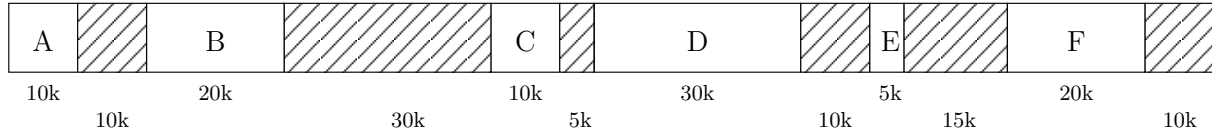


★ Exercice 1. Allocation m moire

On consid re   l'instant t l' tat suivant de la m moire centrale d'un processus :



Des requ tes d'allocation de m moire arrivent dans l'ordre suivant : 20 Ko, 10 Ko, 5 Ko et 25 Ko.

- ▷ **Question 1.**   quelles adresses sont allou s les blocs si on utilise la politique «First Fit»? Quel d cision prise vous semble inefficace?
- ▷ **Question 2.** M me question avec la politique «Best Fit»
- ▷ **Question 3.** M me question avec la politique «Worst Fit»
- ▷ **Question 4.** M me question avec la politique «Next Fit»

★ Exercice 2. Algorithme des fr res siamois.

L'objectif de cet exercice est d' tudier cet algorithme d'allocation m moire. Son principe est de n'alouer que des blocs dont la taille est une puissance de 2. Les blocs disponibles sont rang s dans diff rentes sous-listes en fonction de leur taille et scind s/fusionn s au besoin.

- ▷ **Question 1.** Donnez le pseudo-code de la fonction allouer(). On suppose que le syst me g re une liste pour chaque puissance de i utilis e et on ne demande pas de d tailler la gestion de cette liste (on se limitera    crire par exemple "ins rer l' l ment dans la liste i ").
- ▷ **Question 2.** On suppose que la m moire consiste en un bloc de 256K. Repr senter l' tat d'occupation de la m moire apr s les placements successifs des blocs A(5K), B(25K), C(35K), et D(20K) puis la lib ration des blocs dans l'ordre A, D, C, et B.
- ▷ **Question 3.** Cet algorithme peut  tre g n ralis  en coupant les zones en sous-bloc de tailles diff rentes. Remarquons que pour  viter la fragmentation externe, il faut que les tailles des blocs suivent un syst me de fibonacci.

Donnez l' tat de la m moire apr s application du workload de la question pr c dente en supposant que le d coupage suit les tailles suivantes : 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2594, et 4191.