

Sinfo2 - Allocation Dynamique

19. Fév. 2021

"AVANT"

Tableau à taille fixe → dimension qui connaît les besoins de la composition + simplicité
- PAS SOUPLE

"APRÈS"

Tableau créé lors de l'exécution du programme. - SIMPLE
↳ possibilité d'étendre le tableau si besoin. + GAIN SUR MEMOIRE
Gestion du tableau.

Un fichier txt : "il fait beau"
"le ciel est bleu et la campagne verte".

- Pour transférer le contenu du fichier txt dans une chaîne (qui est un tableau), il nous faut connaître la taille du fichier.
- on peut alors créer un tableau à la bonne dimension

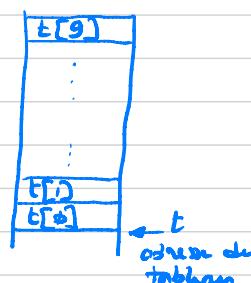
Mode opératoire

- ① Type de données à mettre dans le tableau
- ② Nombre d'éléments dont on a besoin.

Ex: type : double
n : 10

Comparaison tableau statiqu. / dynamique

double t[10];



une variable qui contient l'adresse de début du tableau.

double *t = NULL;

- ③ Demander au système une zone mémoire de la "bonne" taille
la taille est en octets : n * sizeof (double)

t = (double*) malloc (n * sizeof (double));

transtypage (ou cast) : modification explicite du type renvoyé par malloc (void*) par le type attendu pour le tableau.

- ④ Vérifier que l'allocation est correcte

Si erreur t == NULL : NE PAS UTILISER LE TABLEAU !

Si ok t contient l'adresse de début du tableau.

t[0] ... t[n-1] Vous pouvez utiliser le tableau. =)

- ⑤ Après utilisation: libération mémoire (plus besoin du tableau)

free(t);

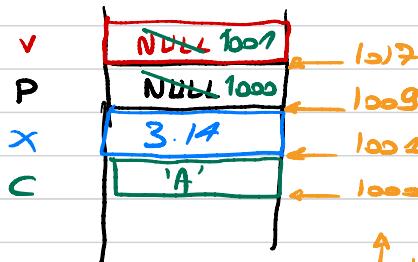
t = NULL; ← précaution importante pour éviter un crash en cas d'utilisation du tableau après sa libération.

Un pointeur est une adresse en mémoire
ou une valeur entière (de 0 à adresse max de la
mémoire disponible)

`char c = 'A';`

`double x = 3.14;`

`char *p = NULL; // p est un pointeur ; il contient l'adresse d'un char`
`double *v = NULL; // v ————— ; ————— double`



↑ les adresses des variables
en mémoire.

`&c : 1000`

`&x : 1001`

`&p : 1009`

`&v : 1017`

des adresses = des entiers (de 0 à la capacité maximale)

`p = &c; // à gauche du "=" pointeur vers un char`
 à droite du "=" adresse d'un char

`v = &x;`

`printf("P = %p"); // P = 1000`
`printf("v = %p"); // v = 1001`

`void *malloc (size_t size);`

↳ renvoie une adresse sur une zone dont
il ne connaît pas le contenu

`int *p = NULL;`

`p = (int *) malloc (200);`

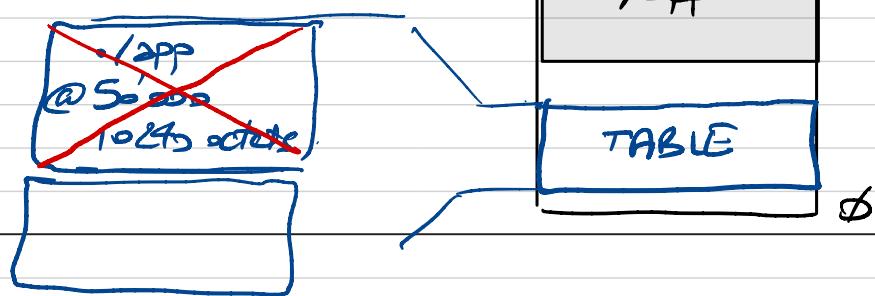
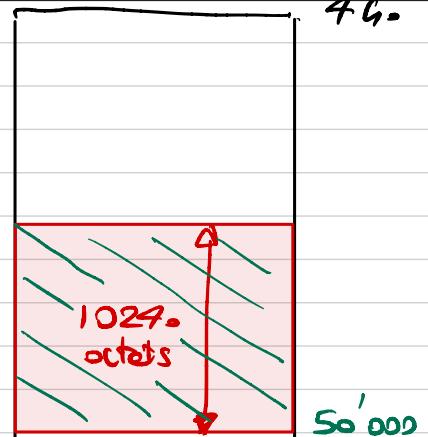
cpp: tableau dynamique de 10 ko.

$p = \underline{\text{malloc}}(1024)$

↳ système,
 $p \leftarrow 5000$

free(p)

46.



TD 2021 04/13

- ① Saisie d'une valeur entière n correspondant au nombre de cases d'un tableau. $\hookrightarrow \text{size_t}$
- ② Allouer dynamiquement un tableau de n cases de type double
- ③ Afficher l'adresse du début de la zone
{ si s/s entier : gerez les erreurs d'allocation.
(msg si allocation incorrecte) }
- ④ Libération de la mémoire allouée

Quelle est la taille limite pour l'allocation (en octets) ?

TD 2021 04/19 b

- ① créer manuellement un fichier texte "info.txt" remplir avec du texte libra (taille comme vous voulez)
<https://fr.lipsum.com/>
- ② Ouvrir le fichier txt et calculer sa taille en octet.
- ③ Allouer un tableau de type char
- ④ Transférer le contenu du fichier vers le tableau.
- ⑤ Analysez le tableau et comptez le nombre de mots. (et affichez).
Séparateur: uniquement l'espace.

