

1 Allocation dynamique de mémoire

L'allocation dynamique de mémoire est généralement nécessaire lors de l'utilisation de structures dont la taille n'est pas connue à la compilation : tableau de taille *a priori* inconnue (par exemple lecture en mémoire de l'intégralité d'un fichier dont on ne connaît pas la taille), utilisation de données de type listes, arbres, graphes...

La fonction d'allocation `malloc` permet d'allouer lors de l'exécution (et non pas lors de la compilation) un nombre donné d'octets. La fonction renvoie un pointeur (de type `void *`) vers la zone allouée. Il est habituel de stocker la valeur renvoyée dans un pointeur typé (par exemple `unsigned int *` si on sait que les données seront des entiers non signés), pour pouvoir utiliser l'arithmétique sur les pointeurs (ajouter 1 au pointeur passera bien à l'entier non signé suivant et pas à l'octet suivant).

La mémoire allouée peut être libérée par la fonction `free` à laquelle on communique un pointeur vers le début de la zone allouée.

La fonction `realloc` prend en paramètre un pointeur vers une zone déjà allouée, une taille en octets, et renvoie un pointeur vers la zone agrandie (qui peut être la même zone, réellement agrandie, ou une nouvelle zone plus grande dans laquelle les données d'origines auront été copiées).

L'exemple suivant alloue une zone mémoire par blocs de 512 octets, et y stocke l'intégralité d'un fichier texte.

```
#include <stdio.h>
#include <stdlib.h>

int main(void) {
    unsigned char * txt;
    size_t taillealloc=512;
    size_t taillereelle=0;
    FILE *f;
    int c;
    txt=malloc(taillealloc*sizeof(char));
    if (txt==NULL) return -1;
    f=fopen("data.txt","r");
    while( (c=fgetc(f))!=EOF ) {
        if (taillereelle>=taillealloc) {
            taillealloc+=512;
            txt=realloc(txt,taillealloc*sizeof(char));
            if (txt==NULL) {fclose(f); return -1;}
        }
        txt[taillereelle]=c;
        taillereelle++;
        //printf("%c",c);
    }
    fclose(f);
    printf("Taille alloc : %ld\n",taillealloc);
    printf("Taille réelle : %ld\n",taillereelle);
    free(txt);
    return 0;
}
```