Programmation et Structures de Données 1995-96
TD 12
Allocation dynamique

Frédéric Vivien, bureau 306, e-mail: Frederic.Vivien

12 janvier 1996

1 Regarder...

Regarder dans le fichier `/usr/include/stdlib.h` la déclaration des fonctions d'allocations dynamiques.

2 malloc

- Allouer dynamiquement un tableau de taille 1000, l'initialiser et afficher les adresses de ses cases extrêmales.
- Écrire une fonction `add_item` qui rajoute un élément à un tableau. Si ce tableau était déjà complet, "l'agrandir" grâce à la fonction `realloc`. Tracer l'adresse de la première case du tableau dans un programme où seront intercalés allocations de tableaux et appels à `add_item` qui engendrent des appels à `realloc`.
- Que fait le programme ci-dessous ? Pourquoi ?

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

char tab[10];

main(){
char * p;
p = tab;
p = (char *) realloc(p, 20);
p[3] = 'a';
printf("%c\n",p[3]);
}
```

  Le programme est disponible soit à l'adresse: `fvivien/Enseignement/Sujets/prog_TD_12.c` soit à l'URL `http://www.ens-lyon.fr/~fvivien/Enseignement/prog_TD_12.c`.

- Stratégie de `malloc`: allouer un certain nombre d'objets (10), en désallouer quelques-uns "au milieu", en ré-allouer. Toujours en traçant les adresses mémoires. Quelle est la stratégie d'allocation de `malloc` ?

3 Chaînes de caractères

- Parcourir, modifier, jouer avec un tableau de caractères défini par: `char s[] = "abc";`
- Écrire une fonction `strcpy` et une fonction `strcat`. Pour cette dernière fonction on supposera que le tableau qui reçoit le résultat de la concaténation est suffisamment grand.

4 Question subsidiaire

Reprendre l'implémentation des listes chaînées du TD10 en n'utilisant plus de tableau `tab` mais en allouant directement les cellules. Seules les fonctions `allocate`, `empty` et `is_empty` devraient avoir besoin d'être modifiées.

Les sources sont disponibles soit dans le fichier `fvivien/Enseignement/Sujets/prog_TD_10.c` soit à l'URL `http://www.ens-lyon.fr/~fvivien/Enseignement/prog_TD_10.c`.