Piles

Fichiers sources : `http://www.labri.fr/perso/mazoit/uploads/2012-algo3-tp1.tgz`

Signature. Les opérations possibles sur une pile sont :
- empiler un objet quelconque ;
- dépiler le dernier objet empilé ;
- récupérer l'objet au sommet de la pile ;
- déterminer si la pile est vide.

Interface. Voilà l'interface C que nous proposons pour une pile (fichier `stack.h`).

```
struct stack_t;
typedef struct stack_t *stack;

/* create an empty stack */
stack stack_create(void);

/* push an object on a stack */
void stack_push(stack s, void *object);

/* return true if and only if the stack is empty */
int stack_empty(stack s);

/* return the top element of the stack. */
void *stack_top(stack s);

/* pop an element off of the stack. */
void stack_pop(stack s);
```

1 Implantation d'une pile avec un tableau extensible

L'objectif de cet exercice est d'implémenter l'interface proposée en utilisant un tableau extensible.

1. On peut imaginer différente stratégie d'extension du tableau lorsque ce dernier devient trop petit pour empiler un nouvel élément. On choisira d'abord la stratégie consistant à étendre la taille du tableau lorsqu'il est plein (ajouter M nouvelles entrées). Vous décrirez d'abord votre implémentation en langage pseudo-code, sans référence aux particularités du langage C. Pour simplifier, vous supposerez en particulier qu'un tableau connaît sa taille (`tableau.taille`).
2. Donnez ensuite une réalisation en C de cette implémentation. (Dans un fichier `stack_array.c`). Tester votre implantation avec le programme de test fourni `test_stack.c`).
3. Générer des cas tests de tailles n croissantes et mesurer le temps avec la fonction `gettimeofday()`. Afficher graphiquement le coût amortie à l'aide de GNUPlot. Vous pouvez utiliser pour cela les fichiers `ALIRE`, `resultat.plot`, `perf_stack.c` fournis dans l'archive.
4. Implémentez la stratégie consistant à doubler la taille du tableau lorsque le tableau est plein.
5. Comparez les résultats obtenus avec les résultats théoriques du cours.

2 Implantation d'une pile avec une liste chaînée

L'objectif de cet exercice est d'implémenter l'interface proposée en utilisant une liste chaînée.

1. Vous décrirez d'abord votre implémentation en langage pseudo-code, sans référence aux particularités du langage C. Discutez des avantages et inconvénients (par rapport à l'implémentation tableau de la section 1).
2. Donnez ensuite une réalisation en C de cette implémentation. (dans un fichier `stack_list.c`). Tester votre implantation avec le programme de test fourni `test_stack.c`).
3. Générer des cas tests de tailles n croissantes et mesurer le temps avec la fonction `gettimeofday()`. Afficher graphiquement le coût amortie à l'aide de GNUPlot. Vous pouvez utiliser pour cela les fichiers `ALIRE`, `resultat.plot`, `perf_stack.c` fournis dans l'archive.

3 Implantation d'une pile avec une « liste chaînée de tableaux »

Reprendre les questions précédentes en implémentant l'interface proposée en utilisant une liste chaîne de tableaux.

Discuter des avantages et des inconvéniants de cette implémentation par rapport aux deux précédents.