Algorithmique et programmation C

Alexandre Guillemot

10 novembre 2022

Table des matières

Introduction

Horaires : 14h00 - 18h15.

Aide mémoire programmation

Compilation

>`gcc [options] fichier.c [arguments] [librairies]`. Les options sont :
- `-E` : préprocesseur uniquement
- `-o nom` : nomme `nom.out` le binaire produit
- `-O, -O1, -O2, -O3` : choix d'efficacité de la compilation, de la pire a la meilleure.
- `-S` : donne le fichier assembleur
- `-W` : donne plus de messages d'avertissement
- `-Wall` : donns tous les messages d'avertissement

>`time ./a.out` pour mesurer le temps d'exécution du programme

Programme et fonctions

structure d'un programme :

```
[directives préprocesseur]
[déclarations variables externes]
[prototype des fonctions externes]
[déclaration des fonctions secondaires]

int main()
{
    instructions
}
```

`#include <lib.h>` (ex `stdio.h`) pour inclure une librarie avec le préprocesseur
template d'une fonction `type fonction(arguments);`
structure d'une fonction

```
type fonction(arguments)
{
    instructions;
```

```
    return var;
}
```

On l'appelle en écrivant `fonction(arguments)`

Types de variables

- Globales : sont stockées sous le tas et est déclarée en dehors du main (donc utilisable par tout le programme)
- Temporaires : variable déclarée à l'intérieur d'une fonction, est détruite en sortie de fonction
- Statiques : `static type var` variable associée à une fonction qui change lorsque l'on appelle cette fonction mais n'est pas détruite en sortie de fonction et conserve sa valeur.

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


//------
typedef struct Polynome {
unsigned int k1;
unsigned int k2;
unsigned int k3;
} polynome;


//------
// Affiche un ui sous forme binaire
void print_bits(unsigned int a);
// Renvoie un polynome primitif de F2^n : F2 sous forme polynome
polynome auxi(unsigned int n);
// Tests de la question 1
void q1();
// Affiche un polynome de F2[T]
void poly_print(unsigned int P);
// Tests de la question 2
void q2();


//------ Fonctions auxiliaires
```

```c
void print_bits(unsigned int a)
{
    for (int i = 0; i < 32; i++)
    {
        printf("%d", (a >> i) & 0x1);
    }
}


//------ Q1
polynome auxi(unsigned int n)
{
    switch (n)
    {
        case 5:
        case 11:
        case 21:
        case 29:
            return (polynome){2, 0, 0};
        case 10:
        case 17:
        case 20:
        case 25:
        case 28:
        case 31:
            return (polynome){3, 0, 0};
        case 9:
            return (polynome){4, 0, 0};
        case 23:
            return (polynome){5, 0, 0};
        case 18:
            return (polynome){7, 0, 0};
        case 8:
        case 19:
            return (polynome){6, 5, 1};
        case 12:
            return (polynome){7, 4, 3};
        case 13:
        case 24:
            return (polynome){4, 3, 1};
        case 14:
```

```
            return (polynome){12, 11, 1};
        case 16:
            return (polynome){5, 3, 2};
        case 26:
        case 27:
            return (polynome){8, 7, 1};
        case 30:
            return (polynome){16, 15, 1};
        default:
            return (polynome){1, 0, 0};
    }
}

unsigned int poly_prim(unsigned int n)
{
    polynome P = auxi(n);
    return 1 ^ (1 << P.k1) ^ (1 << P.k2) ^ (1 << P.k3) ^ (1 << n);
}

void q1()
{
    for (int n = 2; n < 32; n++)
    {
        print_bits(poly_prim(n));
        printf("\n");
    }
}


//------ Q2
void poly_print(unsigned int P)
{
    if (P == 0) printf("0");
    else
    {
        int b = 0;
        int i = 1;
        if (P & 1 == 1)
        {
            printf("%d", 1);
            b = 1;
```

```
        }
        P = P >> 1;
        while (P != 0)
        {
            if (P & 1 == 1)
            {
                if (b == 0)
                {
                    printf("X^%d", i);
                    b = 1;
                }
                else printf(" + X^%d", i);
            }
            i++;
            P = P >> 1;
        }
    }
}

void q2()
{
    for (int n = 2; n < 32; n++)
    {
        poly_print(poly_prim(n));
        printf("\n");
    }
}


//------ Q3



//------
int main()
{
    // q1();
    q2();
}
```