

CC3301 - Programación de Software de Sistemas

Auxiliar 1 Pauta: Conceptos básicos

Profesor: Luis Mateu
Auxiliar: Alexandra Ibarra

13 de Marzo de 2020

1. Compilación

En clases vimos las instrucciones necesarias para compilar, manualmente, un programa escrito en C.

El programa a compilar es *circ.c*, y los pasos para realizarlo son los siguientes:

- **Compilar el archivo *.c* para crear el módulo objeto:**

```
gcc -Wall -c circ.c -o circ.o
```

Esta instrucción crea el módulo *circ.o*, el cual aún no es posible ejecutar pues el sistema operativo no lo reconoce (no sabe como interpretarlo).

Nota: Para que el compilador muestre todos los *warning* que tiene el programa (no son errores, por lo cual compilará pero puede conducir a problemas durante la ejecución), se debe usar el flag *-Wall*.

- **Compilar el archivo *get_pi.c*:** Antes de convertir en ejecutable el módulo compilado anteriormente, se debe compilar el archivo *get_pi.c* a módulo objeto. Esto se debe a que el archivo anterior (*circ.c*) utiliza funciones que están definidas en este otro archivo (*get_pi.c*):

```
gcc -Wall -c get_pi.c -o get_pi.o
```

- **Enlazar ambos módulos en un ejecutable:** Ahora que tenemos todas las dependencias, se puede crear el ejecutable:

```
gcc -Wall -o circ get_pi.o circ.o -lm
```

Nota: Cuando se compila por consola es necesario compilar algunas de las librerías propias de C. En este caso, se debe linkear la librería matemática (*math.h*), por lo cual se utiliza el flag *-lm*.

2. Ejercicios

1)

```
1 void factoresPrimos(int n){
2     int i = 2; // inicio
3     while(i <= n) { // iteramos hasta n incluido
4         if(n%i == 0) { // i es factor de n
5             printf("%d ", i);
6             n = n/i;
7         }
8         else { // i no es factor
9             i++; // pasamos al siguiente
10        }
11    }
12 }
```

```

1 int main(int argc, char *argv[]) {
2     if(argc != 2) {
3         printf("No ingresa cantidad correcta de argumentos\n");
4         return 1; // codigo de error
5     }
6     int n = atoi(argv[1]);
7
8     if(n == 1){
9         printf("1");
10    }
11    factoresPrimos(n);
12    printf("\n"); // salto de linea
13    return 0; // codigo de exito
14 }

```

2 a)

```

1 double llenarMaleta(double w[], double v[], int z[], int n, double maxW, int k) {
2     double best= -1;
3     while (k--) {
4         int x[n];
5         double sumW= 0, sumV= 0;
6         for (int i=0; i<n; i++) { // subconjunto random
7             x[i]= random0or1() && sumW+w[i]<=maxW ? 1 : 0;
8             if (x[i]==1) {
9                 sumW += w[i];
10                sumV += v[i];
11            }
12        }
13        if (sumV>best) { // precio del nuevo subconj es mayor
14            best= sumV; // reemplazamos mejor precio
15            for(int i=0; i<n; i++) {
16                z[i]= x[i]; // reemplazamos el subconjunto
17            }
18        }
19    }
20    return best;
21 }

```

2 b)

```

1 int main(int argc, char **argv) {
2     // no ingresa los parametros correctos
3     if (argc<4) {
4         fprintf(stderr, "Uso: %s <n> <maxW> <k> w0 w1 ... v0 v1 ... \n", argv[0]);
5         exit(1);
6     }
7
8     // n es la cantidad de articulos
9     int n= atoi(argv[1]);
10
11    // si la cantidad de parametros no concuerda
12    if (argc!=4+n+n) {
13        fprintf(stderr, "Uso: %s <n> <maxW> <k> w0 w1 ... v0 v1 ... \n", argv[0]);
14        fprintf(stderr, "argc=%d deberia ser %d\n", argc, 4+n+n);
15        exit(1);
16    }
17
18    double maxW= atof(argv[2]);
19    int k= atoi(argv[3]);
20    double w[n]; // pesos
21    double v[n]; // euros
22
23    for (int i= 0; i<n; i++) {

```

```

24     w[i]= atof(argv[4+i]);
25     v[i]= atof(argv[4+i+n]);
26 }
27 int z[n]; // conjunto elegido
28
29 // calculamos mejor subconjunto
30 double bestSeq= llenarMaleta(w, v, z, n, maxW, k);
31
32 // imprimimos en pantalla el resultado
33 printf("mejor valor= %g\n", bestSeq);
34 for (int i=0; i<n; i++)
35     printf(" %d ", z[i]);
36 printf("\n");
37 return 0;
38 }

```