Departamento de Informática
Sistemas Distribuidos

Ejercicios del laboratorio1

En Aula Global en el apartado Laboratorio1 se tiene disponible este enunciado y el material de apoyo necesario para hacer los ejercicios propuestos. Este material es un fichero `Makefile` que será utilizado para la compilación de los programas que se desarrolle durante este laboratorio.

El objetivo de este laboratorio es familiarizarnos con el lenguaje de programación C.

Ejercicio 1. Escriba un programa (p1.c) que acepte en la línea de mandatos un número indeterminado de argumentos. El programa debe imprimir cada argumento en una línea diferente. El argumento se imprimirá como una cadena de caracteres.

Si el programa se ejecuta de la siguiente forma:

```
./p1 uno 4 hola tres 1245
```

El programa debe imprimir:

```
Argumento 1 = Uno
Argumento 2 = 5
Argumento 3 = Hola
Argumento 4 = Tres
Argumento 5 = 1245
```

Ejercicio 2. Modifique el programa anterior (el nuevo programa será p2.c) asumiendo que los argumentos que se pasan son todos números enteros. El programa debe convertir cada argumento de entrada, cuyo formato es una cadena de caracteres, a un número entero. Para convertir una cadena a un número entero se puede utilizar la función de biblioteca `atoi`:

```
int atoi(const char *nptr);
```

Esta función convierte la cadena apuntada por `nptr` a su valor entero correspondiente. El problema de esta función es que no detecta errores (por ejemplo, cuando se intenta convertir una cadena que no representa un número entero), así que en su lugar podemos utilizar mejor la función equivalente:

```
char *end;
strtol(nptr, &end, 10);
```

En caso de que la conversión no se haya podido realizar, el puntero `end` no habrá avanzado hasta el final. De esta forma se puede detectar el error de la siguiente forma:

```
if (*end != '\0') {
        // error en la conversión
}
```

Si el programa se ejecuta de la siguiente forma:

```
./p1 8 10 23 45
```

El programa debe imprimir:

```
Argumento 1 = 8
Argumento 2 = 10
Argumento 3 = 23
Argumento 4 = 45
```

Los números se imprimirán como enteros. En caso de que el programa se ejecute de la siguiente forma:

```
./p1 8 10 tres 45
```

El programa debe imprimir:

```
Argumento 1 = 8
Argumento 2 = 10
Argumento 3 = Error de conversión.
```

Ejercicio 3. Modifique el programa anterior (el nuevo programa será p3.c) de forma que los argumentos pasados en la línea de mandatos y convertidos a enteros se almacenen en un array creado de forma dinámica. El programa debe incluir una función denominada `ObtenerMinMax()`, que reciba el array y obtenga el valor mínimo y máximo del array.

Ejercicio 4. Modifique el programa anterior (el nuevo programa será p4.c) de forma que los argumentos pasados en la línea de mandatos tratados como cadenas de caracteres se almacenen en un array de forma dinámica. En este caso se creará un array dinámico de cadenas de caracteres.

Ejercicio 5. Modifique los programas anteriores (el nuevo programa será p5.c) de forma que los argumentos pasados en la línea de mandatos se almacenen en una lista enlazada. En cada elemento de la lista se almacenará el valor numérico y el valor como cadena de caracteres. (Véase la guía de C que se encuentra publicada en aula global).

Ejercicio 6. Modifique el programa del ejercicio 3 (el nuevo programa será p6.c) de forma que ordene de menor a mayor los elementos del array utilizando la función de biblioteca `qsort`(). (Véase la guía de C que se encuentra publicada en aula global).