



UNIVERSIDAD  
DE MÁLAGA

Dpto. Lenguajes y  
Ciencias de la Computación

# Programación de Sistemas y Concurrencia Junio 20/6/2013

APELLIDOS \_\_\_\_\_ NOMBRE \_\_\_\_\_

DNI \_\_\_\_\_ ORDENADOR \_\_\_\_\_ GRUPO/TITULACIÓN \_\_\_\_\_

## Ejercicio de C (Bloque 1)

Un sistema operativo usa una lista doblemente encadenada de tipo `T_userList` para gestionar los usuarios del sistema. Un usuario tiene un nombre, un identificador (UID) y un directorio home, de acuerdo a la siguiente definición:

```
typedef struct user {  
    int uid_ ;  
    char *userName_ ;  
    char *homeDirectory_ ;  
  
    struct user * nextUser_ ;  
    struct user * previousUser_ ;  
} T_user ;
```

Definir el tipo `T_userList` con al menos los siguientes campos:  
`head_` y `tail_` (se pueden incluir alguno más si se estima oportuno).  
Las operaciones de la lista son :

`T_user * createUser(char *name, int uid, char *dir);`  
Crea un usuario a partir de su nombre, UID y directorio home.

`T_userList createUserList() ;`  
Devuelve una lista de usuarios vacía

`int addUser(T_userList *, T_user*);`  
Añade un usuario en la cabeza de la lista previa comprobación que ni el nombre ni el UID ya están en la lista. Si se añade con éxito se devuelve 1 y 0 en caso contrario.

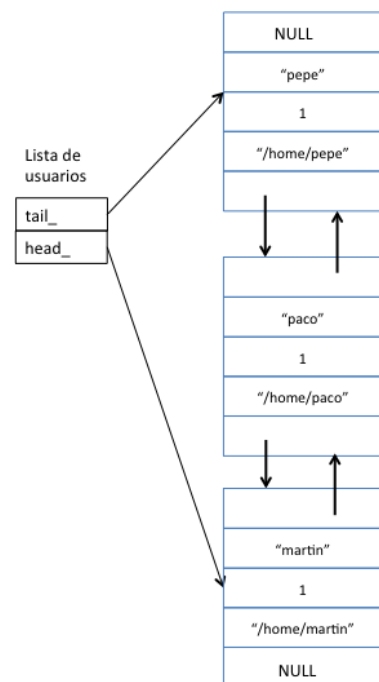
`int getUid(T_userList list, char *userName) ;` Dado el nombre de un usuario, busca su UID en la lista y lo devuelve si lo encuentra; en caso contrario, devuelve -1.

`int deleteUser(T_userList *list, char* userName) ;`  
Borra a un usuario de la lista. Devuelve 0 si la operación tiene éxito y -1 si el usuario no existe.

`void printUserList(T_userList list, int reverse) ;`  
Imprime la lista de cabeza a cola o viceversa dependiendo si el segundo argumento es 0 o 1.

Nota.- Se recomienda utilizar las funciones `strcpy` y `strcmp` de manejo de cadena de caracteres:

```
char *strcpy(char *s1, const char *s2);  
Copia la cadena apuntada por s2 (incluyendo el carácter nulo) a la cadena apuntada por s1
```



```
int strcmp(const char *s1, const char *s2);
```

Compara la cadena apuntada por **s1** con la cadena apuntada por **s2**. **Y**

**Mayor que cero** si  $s1 > s2$  (orden lexicográfico)

**Cero** si  $s1 == s2$  (orden lexicográfico)

**Menos que cero** si  $s1 < s2$  (orden lexicográfico)

A partir del programa principal que se suministra (main.c) se pide implementar los ficheros userList.h y userList.c. La salida por pantalla del programa principal debe ser idéntica a la mostrada en el fichero salida.txt, que también se suministra.

