Programación de Sistemas

Taller Punteros

Introducción

En este taller practicaran el uso de punteros.

Descripción

En este taller, deberán crear un programa sencillo que hará operaciones con punteros. **Deben compilar con las banderas -Wall y -fsanitize=address,undefined.** El programa está organizado en los siguientes archivos:

1. **Funciones.h:** la declaración de las funciones a implementar.
2. **Funciones.c:** Donde implemenetarán sus funciones.
3. **Makefile:** pueden generar su código haciendo **make.** NO modificar el Makefile.
4. **Bytes.c:** no modificar este archivo
5. **Main.c:** aquí puede escribir código para probar sus funciones si desean. NO borrar el código que está después del último comentario.

Las funciones a implementar son:

int parsear_fecha(const char *fecha_str, int *dia, int *mes, int *ano)

Esta función toma un string (fecha_str) con formato día-Mes-Año (por ejemplo, 05-Abr-2021), donde el mes se representa con exactamente 3 letras (la primera mayúscula). La función debe parsear la fecha a su representación numérica. El resultado los devuelve en las variables *dia, *mes y *año. El dia siempre ira desde 00 a 31. TIP: cuando invoquen la función pueden hacerlo así:

```
int dia; int mes; int año;
parsear_fecha(“17-Oct-2018”, &dia, &mes, &año);

//Al retornar las variables dia, mes y fecha tendrán los datos
//parseados
```

Esta función debe **retornar 0 si el parseo fue exitoso, y 1 en caso de error**. En caso de error, las variables dia, mes y año tendrán valor -1. Tienen que validar los siguientes errores:

1. Si cualquiera de los argumentos es NULL
2. Si la fecha tiene cantidad de días inválidos (por ejemplo, 31-Feb-2000).
3. Si la fecha tiene un mes invalido (por ejemplo 15-Acu-2021)

int bytes_significativos(void *valor, unsigned long tamano_valor, unsigned char *bytes, int num_bytes, int mas_menos)

Esta función devuelve los bytes más o menos significativos del **valor.** Si **mas_menos es 0**, se retornan los bytes menos significativos y si es **1**, retorna los más significativos. Los bytes son copiados en el arrego **bytes** que tiene tamaño **num_bytes.** La función también recibe el tamaño de **valor. Asuma maquina LITTLE ENDIAN**. Por ejemplo, si queremos los 3 bytes mas significativos de una variable login haríamos los siguiente:

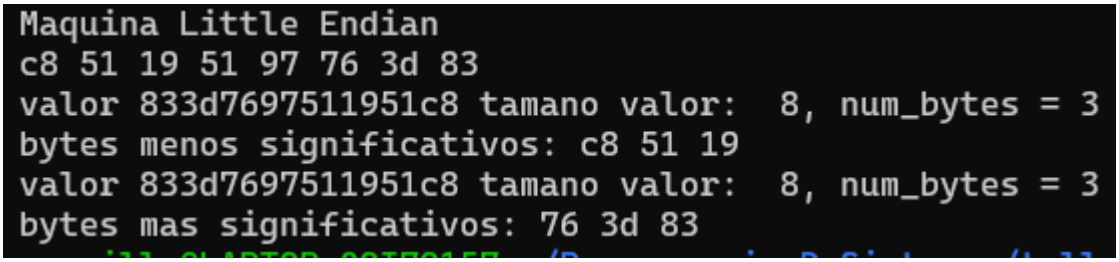
```
int num_bytes = 3;
unsigned char bytes[num_bytes];
long valor = 342432;
bytes_significativos(&valor, sizof(long), bytes, num_bytes, 1);

//al retornar, los bytes mas significativos estarán en el arreglo
//num_bytes
```

Esta función debe **retornar 0 si fue exitoso, 1 si hubo error**. Errores a validar

1. num_bytes debe ser siempre menor o igual a tamaño_valor
2. Si valor o bytes es NULL

Los bytes que vaya encontrando cópielos en el mismo orden que los encuentra en memoria. Un ejemplo de cómo devolver los bytes se muestra abajo:



char *buscar_substring(char *un_string, char *substring)

Buscará el **substring** en el string **un_string**. Devuelve un puntero a la dirección de memoria de la primera letra del substring en **un_string.** Si no lo encuentra, retorna NULL. **NO DEBE USAR LA FUNCION strstr() de C.**

Su función debe validar estos errores (devolver NULL):

1. Si substring es NULL o un_string es NULL