Práctica No. 12

Creación de biblioteca string

Objetivo: Desarrollar aplicaciones de software haciendo uso eficiente de la memoria.

Material:
- Computadora Personal (PC)
- Programa Editor de texto (ASCII), compilador GCC

Equipo:
- Computadora Personal

Desarrollo de la práctica:
Desarrollar una biblioteca para la manipulación de cadenas de caracteres y bloques de memoria. En esta biblioteca queda prohibido el uso de la biblioteca **string.h** de ansi C, así como cualquier otra biblioteca para manipulación de cadena de caracteres.

Las funciones a implementar deberán ser las siguientes:

memcpy
Función que copia los valores de un número definido de bytes desde una locación inicial hasta una locación destino, ambos indicados a través de un apuntador. La copia de los valores se debe realizar de forma binaria.
Prototipo:
```
void * memcpy ( void * destination, const void * source, size_t num );
```

memcmp
Función que compara dos bloques de memoria, acotados por un número de bytes (`num`). La función regresa el valor cero, si ambos bloques de memoria contienen exactamente los mismos datos, regresa un valor menor que cero si el bloque apuntado por `ptr1` es menor al `ptr2`, caso contrario, regresa un valor mayor a cero si el bloque apuntado por `ptr1` tiene un valor mayor a `ptr2`.
Prototipo:
```
int memcmp ( const void * ptr1, const void * ptr2, size_t num );
```

memset
Función que asigna un valor (value) a un bloque de memoria acotado por un determinado número de bytes (`num`).
Prototipo:
```
void * memset ( void * ptr, int value, size_t num );
```

memchr
Función que busca un caracter dentro un determinado bloque de memoria acotado por un determinado número de bytes (num). La función regresa la dirección de memoria en donde se encontró la coincidencia, en caso que no se encuentre coincidencia, regresa null.
Prototipo:
```
void * memchr ( const void * ptr, int value, size_t num);
```

strcpy
Función que copia una cadena de caracteres a otra, incluyendo el caracter null.
Prototipo:
char * strcpy (char * destination, const char * source);

strncpy
Copia los primeros num cantidad de caracteres de la cadena fuente a la cadena destino, si se encuentra el caracter null antes de terminar de copiar los num caracteres, entonces se rellena el espacio restante con ceros.
Prototipo:
```
char * strncpy ( char * destination, const char * source, size_t
num );
```

strcat
Función que concatena una cadena (source) a otra cadena destino (destination).
Prototipo:
```
char * strcat ( char * destination, const char * source );
```

strcmp
Función que compara dos cadenas de caracteres hasta el caracter de fin de cadena. Regresa cero si ambas cadenas son iguales, un número menor a cero si la cadena str1 tiene un valor menor a str2, regresa un número mayor a cero si la cadena str1 contiene un valor mayor a str2.
Prototipo:
```
int strcmp ( const char * str1, const char * str2 );
```

strcmpi (no ansi C)
Función que compara dos cadenas de caracteres hasta el caracter de fin de cadena. Regresa cero si ambas cadenas son iguales, un número menor a cero si la cadena str1 tiene un valor menor a str2, regresa un número mayor a cero si la cadena str1 contiene un valor mayor a str2. Esta función no distingue entre mayúsculas y minúsculas.
Prototipo:
```
int strcmpi ( const char * str1, const char * str2 );
```

strstr
Función que busca una coincidencia de la cadena str2 en la cadena str1, regrese la dirección de memoria que apunta al inicio de la cadena str1 donde se encontró la

coincidencia.
Prototipo:

```
char * strstr (char * str1, const char * str2 );
```

Restricciones

1. En un archivo .c diferente se deben de probar cada una de las funciones desarrolladas.
2. El código fuente deberá contener comentarios que indique el objetivo del programa y descripción de instrucciones más relevantes.
3. Evitar nombres de variables y funciones que no reflejen su propósito claramente.
4. Evitar el uso de biblioteca string.h.

Conclusiones y Comentarios.