DAY #33.

Tuesday, June 30, 2026 8:30 PM

String's functions

Strlen() ⇒ It finds length of string
Strcpy(arg1, arg2) ⇒ It copies arg2 into arg1.

Strncpy(Destination, source, length) ⇒ It copies n characters from source to destination

Strcat(dest, source) ⇒ It is used to concatenate source into destion.

Strncat(dest, source, length) ⇒ It is used to concatenate n number of characters from source to destinat

Strcmp(str1, str2) ⇒ It compare two strings and returns 0 if both are same.

Note:- All of these are predefined function
to use these function we should include header file "string.h"

1. Strlen()

```
Strlen(str);        // It return length of str

char str1[] = "Hello";   OR   char str1[6] = "Hello";
```

'\0' → Null character → string terminator → End of string

Strlen() ⇒ It returns length

```
int len = strlen(str1);   ⇒ 5
printf("length of str1 is %d", len);
```

Dont use extra variable

```
printf("length of str1 is %d", strlen(str1));
```

2. Strcpy(str1, str2);

It will copy str2 into str1.

```
char str2[] = "Hello";
char str1[6];        // 5 for storing "Hello" & 1 for null character.

strcpy(str1, str2);
printf("str1 is %s", str1);
```

str2: H e l l o \0
str1: Garbage value.
H e l l o \0

Hello

Q. What if I print str1 before copying str2 into it ?? ⇒ Undefined behavior OR Garbage values. OR null values

3. Strncpy(dest, source, len)

```
char str1[] = "Hello";   // Suppose I want to copy only 3 char from str1.
char str2[6];  ⇒ "Hel"
Strncpy(str2, str1, 3);
printf("\n str2 is %s", str2);   ⇒ Hel
```

Hello
Hel xxx
Hel xxxxxxx

Better way?

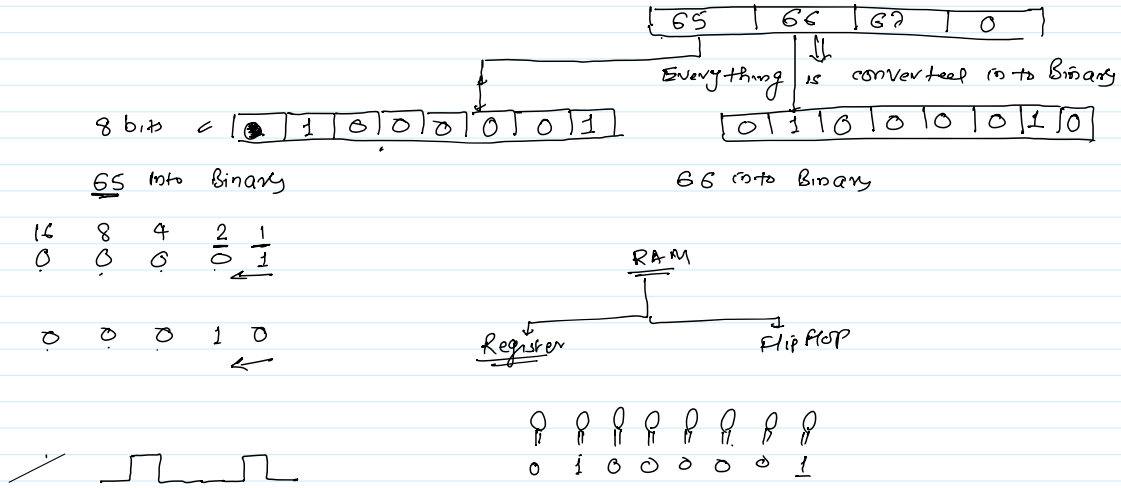
```
char str2[6] = {'\0'};   ⇒ 20%
strcpy(str2, str1, 3);
printf("\n str2 is %s", str2);   ⇒ Hel
```

str2: \0 \0 \0 \0 \0 \0
str2: H e l \0 \0 \0
str1: H e l l o \0

Example. char str1[] = "ABC"; 4B or 4 bytes

⇒ char str1[] = {65, 66, 67, 0};

'A' 'B' 'C' '\0'


65 66 67 0
Everything is converted into Binary
8 bits =
65 into Binary
66 into Binary
64 32 16 8 4 2 1
RAM
Register
Flip Flop