strlen

Syntax: size_t strlen(const char *s);

- DESCRIPTION
- The strlen() function calculates the length of the string pointed to by s, excluding the ter-minating null byte ('\0').

- RETURN VALUE
- The strlen() function returns the number of bytes in the string pointed to by s.

strlen

h	e	l	l	o		w	o	r	l	d	w	j	u	h	f	w	\0		
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17		

0x1000

- For(i=0;str[i];i++);
- here initially i value is 0 loop will end when i becomes 17 because of '\0'.
- so here i value 17 is length of the string. Excluding '\0' character and if you include '\0' character strlen will be 18 because 0 -17 means 18.

Strcpy

syntax: char *strcpy(char *dest, const char *src);

- DESCRIPTION
- The strcpy() function copies the string pointed to by src, including the terminating null byte('\0'), to the buffer pointed to by dest.
- RETURN VALUE

The strcpy() function return a pointer to the destination string dest.

Strcpy

- Char src[20]="hello world";
- Char dest[20];

h	e	l	l	o		w	o	r	l	d	\0								
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

src

- Strcpy(dest,src);

h	e	l	l	o		w	o	r	l	d	\0								
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

dest

- And at last '\0' character also got added.

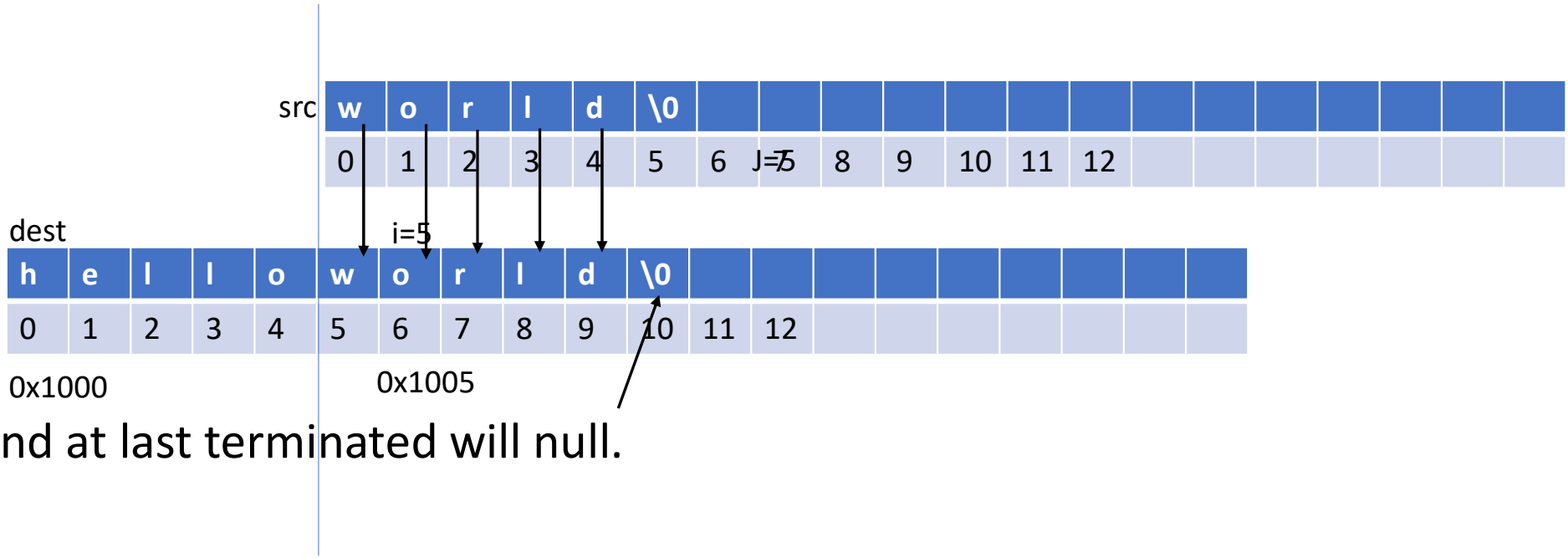
Strcat

syntax: char *strcat(char *dest, const char *src);

- DESCRIPTION
- The strcat() function appends the src string to the dest string,overwriting the terminating null byte ('\0') at the end of dest, and then adds a terminating null byte. The strings may not overlap, and the dest string must have enough space for the result. If dest is not large enough, program behavior is unpredictable.
- RETURN VALUE
- The strcat() functions return a pointer to the resulting string dest.

strcat

- Char src[10]="world";
- Char dest[20]="hello";



And at last terminated will null.

strchr

Syntax: Char *strchr(const char *,int c);

DESCRIPTION

- The strchr() function returns a pointer to the first occurrence of the character c in the string s.

RETURN VALUE

- The strchr() function return a pointer to the matched character or NULL if the character is not found. The terminating null byte is considered part of the string, so that if c is specified as '\0', these functions return a pointer to the terminator.

Char str[20]="hello worldwjuhfw" ;char ch='w';

h	e	l	l	o		w	o	r	l	d	w	j	u	h	f	w	\0		
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16			

0x1000

'w'=='h' False

w

h	e	l	l	o		w	o	r	l	d	w	j	u	h	f	w	\0		
0	1	2	3	4	5	6	7	8	9	10	11	12							

0x1001

'w'=='e' False

w

h	e	l	l	o		w	o	r	l	d	w	j	u	h	f	w	\0		
0	1	2	3	4	5	6	7	8	9	10	11	12							

0x1002

'w'=='l' False

w

Char str[20]="hello worldwjuhfw" ;char ch='w';

h	e	l	l	o		w	o	r	l	d	w	j	u	h	f	w	\0		
0	1	2	3	4	5	6	7	8	9	10	11	12							

0x1003

'w'=='l' False

w

h	e	l	l	o		w	o	r	l	d	w	j	u	h	f	w	\0		
0	1	2	3	4	5	6	7	8	9	10	11	12							

0x1004

'w'=='o' False

w

h	e	l	l	o		w	o	r	l	d	w	j	u	h	f	w	\0		
0	1	2	3	4	5	6	7	8	9	10	11	12							

0x1005

'w'==' ' False

w

Char str[20]="hello worldwjuhfw" ;char ch='w';

h	e	l	l	o		w	o	r	l	d	w	j	u	h	f	w	\0		
0	1	2	3	4	5	6	7	8	9	10	11	12				16			

0x1006

'w'=='w' True return 0x1006

w

So here it return pointer to first occurrence of a character 'w' in string str.

Char str[20]="hello hi" ;char ch='w'; Another case if char not present

h	e	l	l	o		h	i	\0											
0	1	2	3	4	5	6	7	8	9	10	11	12							

0x1000

w

'w'=='h' False

h	e	l	l	o		h	i	\0											
0	1	2	3	4	5	6	7	8	9	10	11	12							

0x1001

'w'=='e' False

w

h	e	l	l	o		h	i	\0											
0	1	2	3	4	5	6	7	8	9	10	11	12							

0x1002

'w'=='l' False

w

Char str[20]="hello hi" ;char ch='w';

h	e	l	l	o		h	i	\0											
0	1	2	3	4	5	6	7	8	9	10	11	12							

0x1003

'w'=='l' False

w

h	e	l	l	o		h	i	\0											
0	1	2	3	4	5	6	7	8	9	10	11	12							

0x1004

'w'=='o' False

w

h	e	l	l	o		h	i	\0											
0	1	2	3	4	5	6	7	8	9	10	11	12							

0x1005

'w'==' ' False

w

Char str[20]="hello hi" ;char ch='w';

h	e	l	l	o		h	i	\0											
0	1	2	3	4	5	6	7	8	9	10	11	12							

0x1006

w

'w'=='h' False

h	e	l	l	o		h	i	\0											
0	1	2	3	4	5	6	7	8	9	10	11	12							

0x1007

'w'=='i' False

w

h	e	l	l	o		h	i	\0											
0	1	2	3	4	5	6	7	8	9	10	11	12							

0x1008

w

'\0' so return NULL

Not found returns NULL.

strrchr

Syntax: char *strrchr(const char *s, int c);

DESCRIPTION

The strrchr() function returns a pointer to the last occurrence of the character c in the string s.

RETURN VALUE

The strchr() and strrchr() functions return a pointer to the matched character or NULL if the character is not found. The terminating null byte is considered part of the string, so that if c is specified as '\0', these functions return a pointer to the terminator.

Char str[20]="embedded" ;char ch='e';

e	m	b	e	d	d	e	d	\0											
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16			

0x1000

e

'e'=='e' true

p=0x1000

e	m	b	e	d	d	e	d	\0											
0	1	2	3	4	5	6	7	8	9	10	11	12							

0x1001

'm'=='e' False

e

e	m	b	e	d	d	e	d	\0											
0	1	2	3	4	5	6	7	8	9	10	11	12							

0x1002

'b'=='e' False

e

Char str[20]="embedded" ;char ch='e';

e	m	b	e	d	d	e	d	\0											
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16			

'e'=='e' true

e p=0x1003

e	m	b	e	d	d	e	d	\0											
0	1	2	3	4	5	6	7	8	9	10	11	12							

0x1004

'd'=='e' False

e

e	m	b	e	d	d	e	d	\0											
0	1	2	3	4	5	6	7	8	9	10	11	12							

0x1005

'd'=='e' False

e

Char str[20]="embedded" ;char ch='e';

e	m	b	e	d	d	e	d	\0											
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16			

'e'=='e' true

e p=0x1006

e	m	b	e	d	d	e	d	\0											
0	1	2	3	4	5	6	7	8	9	10	11	12							

0x1007

'd'=='e' False

e

e	m	b	e	d	d	e	d	\0											
0	1	2	3	4	5	6	7	8	9	10	11	12							

0x1008

Return p

e

Strrev

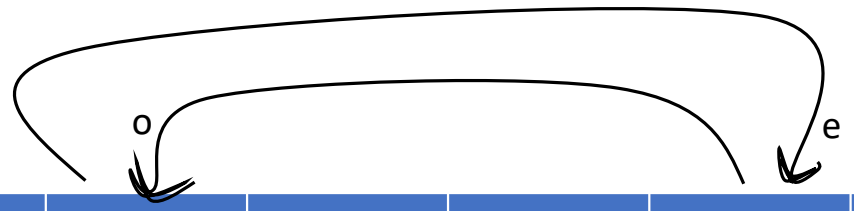
syntax: Char * strrev(char * str);

- input: “vector” “vector chennai”
- Output: “rotcev” “iannehc rotcev”

v	e	c	t	o	r	\0
0	1	2	3	4	5	6

i=0 < J=5

Strrev

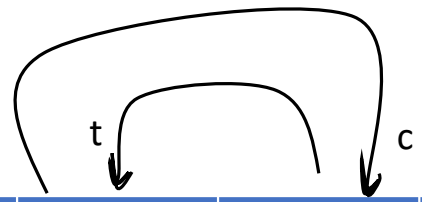


r	e	c	t	o	v	\0
0	1	2	3	4	5	6

i=0 < J=5

i=1 < J=4

Strrev



r	o	c	t	e	v	\0
0	1	2	3	4	5	6

i=0 < J=5

i=1 < J=4

i=2 < J=3

Strrev

r	o	t	c	e	v	\0
0	1	2	3	4	5	6

i=0 < J=5

i=1 < J=4

i=2 < J=3

i=3 < J=2

F

- So finally i<j --> 3<2 false loop will fail.
- And string is reversed.