C under Linux

Dr. Naeem Odat

Department of Computer and Communications Engineering

C - Strings

C - Strings

C - Strings

Strings in C

- There are no string data type in C.
- Part of the convention includes terminating character array with a null character.

```
#include <stdio.h>
int main () {
    char str[6] = {'H', 'e', 'l', 'l', 'o', '\0'};
    //char str[]="Hello";
    printf("The message is: %s\n", str );
    return 0;
}
```

- In many cases string operations in C look just like those found in other languages.

C - Strings

String Operations - *man string*

```
#include<string.h>
char *strcat(char *dest, const char *src);
    Append the string src to the string dest, returning a pointer dest.

char *stpcpy(char *dest, const char *src);
    Copy a string from src to dest, returning a pointer to the end of the resulting string
    at dest.

char *strchr(const char *s, int c);
    Return a pointer to the first occurrence of the character c in the string s.

int strcmp(const char *s1, const char *s2);
    Compare the strings s1 with s2.

char *strcpy(char *dest, const char *src);
    Copy the string src to dest, returning a pointer to the start of dest.

size_t strlen(const char *s);
    Return the length of the string s.

char *strtok(char *s, const char *delim);
    Extract tokens from the string s that are delimited by one of the bytes in delim.
```

C - Strings

String Concatenation

- Example:

```
#include <stdio.h>
#include <string.h>
int main(){
    char *a = "Hello ";
    char *b = "world";
    strcat(a, b);
    printf("%s\n", a);
    return 0;
}
```

- Constant pointers can not be changed.

C - Strings

String Concatenation

- Example:

```
#include <stdio.h>
#include <string.h>
int main() {
    char a[12] = "Hello ";
    char *b = "world";
    strcat(a, b);
    printf("%s\n", a);
    return 0;
}
```

- Make sure the destination array has enough space.

C - Strings

String Copy

```
#include<stdio.h>
#include<string.h>
int main(){
    char s1[15];
    strcpy(s1, "Hello World");
    printf("The length of %s is: %d\n", s1, strlen(s1));
    ...
    return(0);
}
```

C - Strings

String Tokenizing

```
#include <stdio.h>
#include <string.h>
int main(){
    char str[] = "I am an AI Engineer";
    // Returns first token
    char* token = strtok(str, " ");
    // Keep printing tokens while one of the
    // delimiters present in str[].
    while (token != NULL){
        printf("%s\n", token);
        token = strtok(NULL, " ");
    }
    return 0;
}
```