Text files working with strings
include <string.h>

Program Persistent Data

Lecture 3

Review

- In C there are buffers required to work with files.
- Streams are declared using **`FILE *fp;`**
- These streams are required for each file that you work on.
- To open and use the stream, error check that the file exists then close when finished:

```
fp = fopen("write.txt","w");
if (fp == NULL)
     {printf("Can't open file.\n");}
fclose(fp);
```

Review – *text* file <stdlib.h>

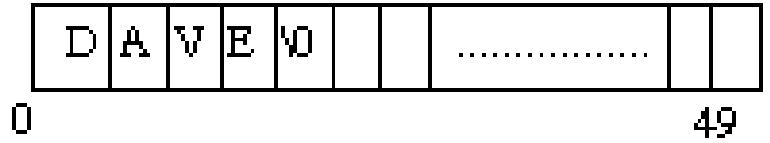
Instruction	Meaning
`fgetc(fp)`	Read a char from file using stream.
`fputc(fp)`	Write a char to file using stream.
`fgets(string,size,fp)`	Read a string from file using stream. It reads a string of a specified size.
`fputs(string,fp)`	Write a string to file using stream.
`fprintf(fp,"Hi %s, you are %i",s,a)`	Write the content to the file using stream.
`fscanf(fp,"%s %s %i",a,b,&c)`	Read a formatted line from file using stream.

#include <string.h>

String Library

What do we know about this library?
What do we want to do with strings??

What do we know about strings?



- Strings in C are arrays of chars
- Must end with termination *null* character '\0'

/*otherwise C does not know when the string is over*/

- Use %s to print a string (well).

```c
/*How to print out a string*/
/*print hi but need 1 extra for terminating char \0*/
#include <stdio.h>
main()
{char str[3];
str[0]='h';
str[1]='i';
str[2]='\0';

/*Use %s to stdout a string*/
printf("%s\n", str);
}
```

Review: strings in C

Faster way to initialize a string when the string is declared:

```
char text[10]="my text"; //this is a dynamic array
```

Careful now: C will allocate only space to fit the initial value of the string; it is not possible to assign a longer text to it!

Char pointers

Strings cannot be assigned a value in this way:

```
text = "new text";    //compiler error
```

BUT can be via the char pointer

```
Char *text;

text = "new text";
```

Careful now: C will create a 'string literal', this can't be altered.

So you can not use `scanf()` with a char pointer.

#include <string.h>

Name	Example	Meaning
1. Strlen()	`len=strlen(str);`	Get the length of a string
2. strcmp	`strcmp(str, "jane" )`	Compare 2 strings
3. strcpy	`strcpy(str, "jane");`	Copy a strings to another
4. strcat	`strcat( string, " Ferris" );`	Concatenate 2 strings
5. strstr	`Strstr(str, "jane")`	Look for a substring in a string

Strlen()

```c
/*function strlen*/
/*print hi but need 1 extra for terminating char \0*/
#include <stdio.h>
main()
{int i;
char str[3];
str[0]='h';
str[1]='i';
str[2]='\0';
/*Use %s to stdout a string*/
printf("%s\n", str);
i=strlen(str);
printf("Length: %i \n", i);
}
```

Strcmp()

```c
/*function strcmp*/
#include <stdio.h>
main()
{int i;
char str[3];
str[0]='h';
str[1]='i';
str[2]='\0';
printf("%s\n", str);
/*Use fn strcmp to return int*/
/*1 higher; -1 lower & 0 same*/
i=strcmp(str, "abc");
printf("Outcome: %i \n", i);
}
```

Strcpy()

```c
/*function strcpy*/
#include <stdio.h>
#include <string.h>
main()
{char str[3];
/*Use fn strcpy to copy the string value*/
strcpy(str,"hi");
printf("%s \n", str);
}
```

Strcpy()careful

```
#include <stdlib.h>
#include <stdio.h>
int main()
{       int i;
        char str[3];

        strcpy(str,”Hi”); //copy in content using strcpy
        printf("%s\n", str);   // prints to stdout

        char str2[10];////but copying strings be Careful
        //that the other is correct size
        strcpy(str2, str);
        printf ("Value of str2: %s \n",str2);
getchar();
return 0;
}
```

Strcat()

```c
/*function strcat*/
#include <stdio.h>
#include <string.h>
main()
{int i;
char str[3];
str[0]='h';
str[1]='i';
str[2]='\0';
printf("%s \n", str);
/*Use fn strcat to join 2 strings*/
char str2[10];
strcpy(str2,"say ");
strcat(str2, str);
printf("Joining the 2 words: %s \n", str2);
}
```

Strstr()

```c
/*function strstr*/
#include <stdio.h>
#include <string.h>
main()
{int i;
char str[3], str2[10];
str[0]='h';
str[1]='i';
str[2]='\0';
printf("%s \n", str);
strcpy(str2,"say hi");
/*Use fn strstr to check if a substring*/
if (strstr(str2, str)==NULL)
{printf("Not found \n");}
else
{printf("found \n");}
}
```

lab2

- Working within the teams.
- Working with the text files, stdlib.h & string.h functions:
- Two tasks:

1. Merge 2 .txt files.
2. Using strstr to search for the string ‘program’ within a text file.
3. Using fscanf with a file given in webcourses.