

13.6 Standard input e standard output

Quando un programma va in esecuzione il sistema apre automaticamente tre file pointer, mediante i quali è possibile scrivere messaggi a video e acquisire dati dalla tastiera. Questi tre file pointer prendono il nome di Standard Input (`stdin`), Standard Output (`stdout`) e Standard Error (`stderr`) e possono essere utilizzati dalle funzioni di accesso ai file descritte nei precedenti paragrafi.

Il file pointer `stdin` è associato dal sistema alla tastiera, i due file pointer `stdout` e `stderr` sono entrambi assegnati al video. Per scrivere un messaggio a video si può allora utilizzare, per esempio, la funzione `fprintf`:

```
#include <stdio.h>
main()
{
    fprintf(stdout, "Ciao lettore\n");
}
```

Dunque le funzioni che abbiamo utilizzato per accettare valori da tastiera e mandare uscite al video corrispondono a usi particolari delle funzioni di uso generale esaminate in questo capitolo. Si hanno le seguenti equivalenze:

<code>printf(...)</code>	->	<code>fprintf(stdout, ...)</code>
<code>scanf(...)</code>	->	<code>fscanf(stdin, ...)</code>
<code>getchar()</code>	->	<code>fgetc(stdin)</code>
<code>putchar(c)</code>	->	<code>fputc(c, stdout)</code>
<code>eof()</code>	->	<code>feof(stdin)</code>

La sintassi delle funzioni a sinistra è più sintetica, perché quelle sulla destra devono specificare che desiderano operare sullo standard input o sullo standard output.

Il programmatore deve prestare molta attenzione all'utilizzo delle due funzioni con sintassi abbreviata `gets` e `puts`, il cui comportamento è simile ma non uguale a quello delle funzioni `fgets` e `fputs`. Infatti `gets` legge una riga da tastiera ma elimina il carattere di newline e `puts` scrive una riga a video aggiungendo automaticamente un carattere di newline. Ne consegue che :

<code>gets(buf, n)</code>	non equivale a	<code>fgets(buf, n, stdin)</code>
<code>puts(buf)</code>	non equivale a	<code>fputs(buf, stdout)</code>