

## Argv, argc e alcune importanti osservazioni sulle stringhe

Vettore argv[]

In ambiente MSDOS per mandare in esecuzione un programma **prog.exe** l'utente digita il nome del programma (l'estensione .exe può essere omessa) eventualmente seguito da uno o più **parametri**.

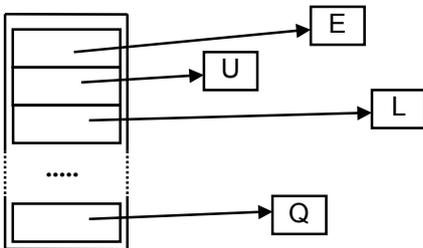
Poichè ai fini pratici l'attivazione di un programma C significa iniziare l'esecuzione chiamandone la funzione **main(int argc, char \* argv[])**, funzione che richiede due parametri, MSDOS per attivare un programma C dovrà sapere quale valore dare a tali parametri all'atto della chiamata della funzione.

Il valore del primo parametro, **argc**, è pari al numero di elementi contenuti nel vettore indicato dal secondo parametro. Vediamo dunque nel dettaglio il secondo parametro, **argv**.

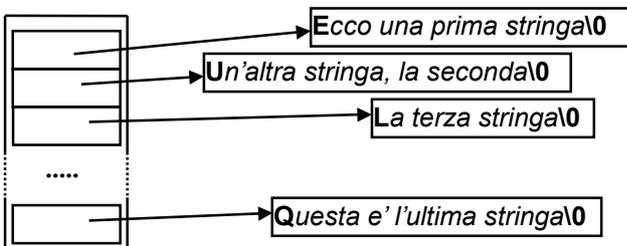
Come risulta dalla dichiarazione, si tratta di **un array di (un numero imprecisato di) char \***,



ossia **un array di punteggiatori**, ognuno dei quali punta a **un carattere**.



**char \*** è, a rigore, un puntatore a un carattere, ma se si fa in modo tale che il carattere a cui punta non sia altro che **il primo carattere di una stringa**,



*Per una spiegazione del ruolo del carattere \0 presente in fondo alle stringhe si veda più avanti.*

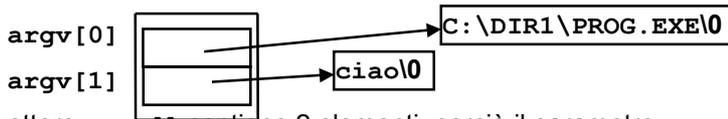
si intuisce che passare tale informazione è sufficiente perchè il ricevente possa accedere all'intera stringa, percorrendola "in avanti" a partire dal suo primo carattere. Questo è infatti il modo standard per passare stringhe in C.

In base a questa interpretazione, risulta chiaro che il parametro argv della funzione main di un programma serve per contenere delle stringhe. Quante e quali sono queste stringhe?

- **La prima stringa contenuta nell'array argv, ossia argv[0]** (ricordarsi che le posizioni negli array si numerano a partire da 0 e non da 1), è **sempre presente** e contiene il path completo del programma che si sta eseguendo. Poichè nell'array **argv** è sempre presente almeno questa prima stringa, il parametro **argc**, che contiene il numero di parametri contenuti in argv, **varrà sempre almeno 1**.
- Le stringhe successive (ossia argv[1], argv[2], e così via) sono presenti soltanto se sono stati forniti parametri sulla riga di comando. Ogni parametro presente sulla riga di comando, nell'ordine da sinistra verso destra, viene inserito in una di queste stringhe. Esempio:

Supponiamo di trovarci nella directory **c:\dir1** nella quale si trova il programma **prog.exe** e di aver impartito il seguente comando: **prog ciao**

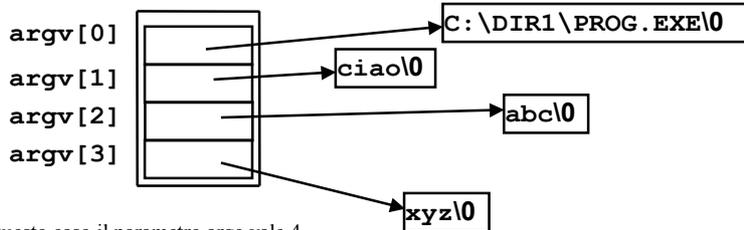
Ecco che cosa conterrà il vettore **argv[]** in questo caso:



Il vettore `argv[]` contiene 2 elementi, perciò il parametro `argc` vale 2.

Se il comando fosse stato `prog ciao abc xyz`, la situazione sarebbe stata la seguente:

Vettore `argv[]`

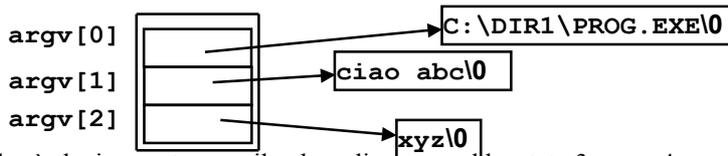


In questo caso il parametro `argc` vale 4.

MSDOS interpreta la riga di comando "spezzandola" laddove sono presenti spazi. I "pezzi" così ottenuti sono usati: il primo come nome del programma; il secondo e successivi, ove presenti, come parametri. Se non esistessero modi per modificare questo comportamento, non sarebbe mai possibile avere un parametro contenente spazi.

Utilizzando le virgolette è possibile raggruppare le parole presenti sulla riga di comando, facendo in modo che MSDOS la interpreti (e la spezzi) diversamente: con riferimento all'ultimo esempio, se la riga di comando fosse stata `prog "ciao abc" xyz`, ecco che cosa sarebbe successo:

Vettore `argv[]`



Va da sé che in questo caso il valore di `argc` sarebbe stato 3 e non 4.