

Paso de Argumentos desde la Línea de Comandos

Un programa puede recibir argumentos desde la línea de comandos, es decir, al momento de pedir a la computadora desde la terminal que ejecute un programa con el comando:

```
./Programa.out
```

A este comando se le pueden pasar argumentos de la siguiente manera:

```
./Programa.out 1 2 3 4
```

Para decirlo con otras palabras, esto sustituye a un `scanf` dentro del programa. Esta característica guarda los argumentos pasados por el usuario desde la línea de comandos en un arreglo de tamaño dinámico (cambia dependiendo de la necesidad). También el programa crea y modifica una variable adicional que detecta cuantos fueron los argumentos pasados a un programa desde la línea de comandos. Para indicarle a un programa que le vamos a pasar argumentos desde la línea de comandos se efectúa el siguiente cambio:

Función main normal:

```
int main(void)
```

Función main con argumentos:

```
int main(int argc, char *argv[])
```

A diferencia de la definición de `main` en la cual la función no recibe nada, cuando le indicamos que le vamos a pasar argumentos desde la línea de comandos, ponemos como referencia las variables `int argc`, `char *argv[]`. Donde `argc` es el número de argumentos que se le pasan al programa desde la línea de comandos, y `*argv[]` es donde se guardan los argumentos. El problema de esto es que los argumentos quedan guardados como caracteres dentro de un arreglo de tamaño variable. Para cambiar el tipo de dato hay que usar la función `sscanf`, que de cierta forma funciona como conversor, y se utiliza de la siguiente manera:
`sscanf(argv[i], "%d", &j);`, donde `argv[i]` es de dónde va a leer, `"%d"` es el tipo de dato que debería de ser (de otra forma al tipo de dato al que se desea convertir) y `&j` es la variable a donde se almacena.