

LOS ARGUMENTOS DE LOS PROGRAMAS

Horacio Goetendía Bonilla

26 de julio de 2003

1. Convenciones sintácticas de los argumentos

POSIX recomienda las siguientes convenciones para los argumentos pasados por línea de comando, las cuales son usadas por las funciones `getopt` y `argp_parse` de la librería GNU de C, pero antes de pasar a usar estas funciones recordemos las convenciones del uso de opciones y argumentos dictadas por POSIX.

- Los argumentos son opciones si comienzan con el signo delimitador (-) ¹.
- Múltiples opciones pueden seguir en un mismo signo delimitador(-) en un simple grupo o bloque si las opciones no van a tomar argumentos. Esto es `-abc` es igual a `-a -b -c`, otro ejemplo es el comando `ls` con las opciones `ls -la` las cuales listan todos los archivos en modo completo, estas opciones no esperan argumentos por eso son susceptibles a ser agrupadas.
- Los nombres de las opciones deben ser siempre caracteres alfanuméricos.
- Ciertas opciones requieren de argumentos. Por ejemplo la opción `-o` del comando `gcc` requiere un argumento (non-option argument) que en este caso es el nombre del archivo salida de la compilación.
- Una opción y su argumento podrían o no estar unidos. Por ejemplo `-omiarchivo` es equivalente a `-o miarchivo`.
- Las opciones típicamente preceden otros non-option arguments.

La implementación de `getopt` y `argp_parse` en la librería de C de GNU normalmente dan la apariencia de como si todas las opciones y argumentos fueron especificados antes de todos los non-option arguments para el propósito de describirlo gramaticalmente, igual que si el usuario de tu programa intercalase opciones y non-option arguments. Estas funciones hacen esto por reordenar los elementos del array `argv`. Este comportamiento no está estandarizado; si se desea eliminarlo definimos la variable de entorno `_POSIX_OPTION_ORDER`.

- El argumento `--` termina todas las opciones; cualquier argumento que siga es tratado como non-option arguments, igual que si ellos empiecen con un hyphen (dash).

¹Usualmente llamado dash

- Un grupo o bloque que consiste de un simple hyphen (dash) es interpretado como un ordinario non-option argument. Por convención, esto es usado para especificar entrada de o salida a la entrada estandar y flujos de salida.
- Las opciones podrian ser llamadas en cualquier orden o aparecer varias veces. Por ejemplo `ls -la` es igual a `ls -al`.

GNU a aumentado *long options* a estas convenciones. Las *long options* consisten de ‘-’ seguidas por un nombre de caracteres alfanuméricos y dashes. Los nombres de las opciones son tipicamente de una a tres palabras de longitud con hypens para separar las palabras. Los usuarios pueden abreviar los nombres de las opciones tan largo como la abreviación sea única.

Para especificar un argumento para un *long option*, escribimos ‘-nombre=valor’. Esta sintaxis habilita un *long option* para aceptar un argumento que es por si mismo opcional.

Eventualmente, el sistemas GNU proveerá complementos para los nombres *long option* en el *shell*.

2. Pasando argumentos a los programas

Si la sintaxis para los argumentos de la linea de comandos para tu programa es simple, puedes simplemente tomar los argumentos de `argv` a mano. Pero si tu programa toma un razonable número de argumentos o todos los argumentos son interpretados de la misma manera (como archivos por ejemplo) puedes optar por usar `getopt` o `argp_parse` para hacer el paso de argumentos.

`getopt` es más estandarizado (estandar POSIX), pero `argp_parse` es con frecuencia mucho más fácil de usar, ambos hacen el trabajo pesado en el traspaso de los argumentos por usted.

3. Conjugando opciones usando getopt

Las funciones `getopt` y `getopt_long` automatizan el paso de las opciones y argumentos.

3.1. Usando la función getopt

Acá se detalla acerca de como llamar a la función `getopt`. Para usarlo tu programa debe incluir la cabecera ‘`unistd.h`’.

Variables globales de getopt

int opterr Si este valor es diferente de cero entonces `getopt` imprimi6n un mensaje de error a `stderr`, este error se emite si se encuentra una opci6n desconocida o una opci6n que no tiene un argumento requerido. Este es el comportamiento por defecto. Si *seteas* esta variable a cero, `getopt` no imprimirá ning6n mensaje, pero se permanecerá retornando el caracter ‘?’ para indicar el error.

int optopt Cuando `getopt` encuentra una opci6n desconocida o una opci6n que no tiene un rgumento requerido, esta variable guarda el caracter de la opci6n en esta variable. Puedes usarlo para proveer tus propios mensajes de diagn6stico.

int optind Esta variable es *seteada* por **getopt** a el indice de el siguiente elemento del array de **argv** para ser procesado. Una vez que **getopt** a encontrado todos los opciones y argumentos, puedes usar esta variable para determinar donde empiezan en resto de non-option arguments. El valor inicial de esta variable es 1.