Argumentos para el main

Introducción

Cuando uno normalmente ejecuta un proceso C, lo realiza a través de la línea de comandos de una terminal.

Por ejemplo, si tomamos el cliente del TP0:

```sh
./Debug/client
```

Sin embargo, uno a veces necesita parametrizar algunos valores para que difieran entre una ejecución y otra. Ahí entran en juego los archivos de configuración que realizamos en el TP0: para parametrizar, por ejemplo, la IP en la que se encontraba el servidor.

Pero, ¿y si quisiera mantener muchos archivos de configuración para distintas ejecuciones? 🤔

Podría tener varios archivos y ponerlos en la carpeta en donde mi proceso espera encontrarlos, pero eso se vuelve engorroso y difícil de mantener, porque, si levantamos varias instancias con distintas configuraciones, no es posible determinar qué contenía ese archivo al arrancar a ejecutar cada una.

Ahí es donde pueden resultarnos muy útiles los **parámetros del main**, para así poder pasarle la ruta al archivo que queramos (y cualquier otra cosa también).

argc y argv con un ejemplo

Entonces, si modificamos al cliente del TP0 para que tome una ruta por `main()` podemos hacerlo de la siguiente manera:

c

```c
int main(int argc, char** argv) {

    //resto del TP0 de antes

    t_config* config = crear_config(argv[1]);

    //resto del TP0 de después
}
```

Y lo ejecutamos como:

sh

```sh
./Debug/cliente ./una/ruta/a/mi/archivo.cfg
```

- `argc` es la cantidad de argumentos que se agregan por línea de comando (**arg**ument **c**ount).
- `argv` es un array de strings que contiene los string ingresados (**arg**ument **v**ector).

El motivo por el que el segundo elemento del array es la ruta que ingresamos es porque el primer elemento es siempre el comando en sí mismo (en este caso, `./Debug/cliente`).

Esto incluso lo podemos mejorar controlando que la cantidad de parámetros sea la indicada manejando `argc`:

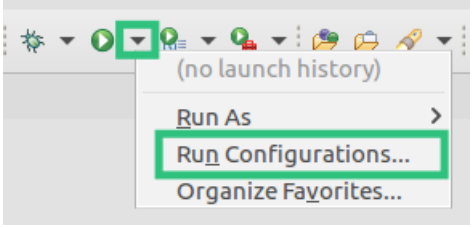
c

```c
int main(int argc, char** argv) {
    if (argc < 2) {
        return EXIT_FAILURE;
    }

    //resto del TP0 de antes

    t_config* config = crear_config(argv[1]);

    //resto del TP0 de después
}
```

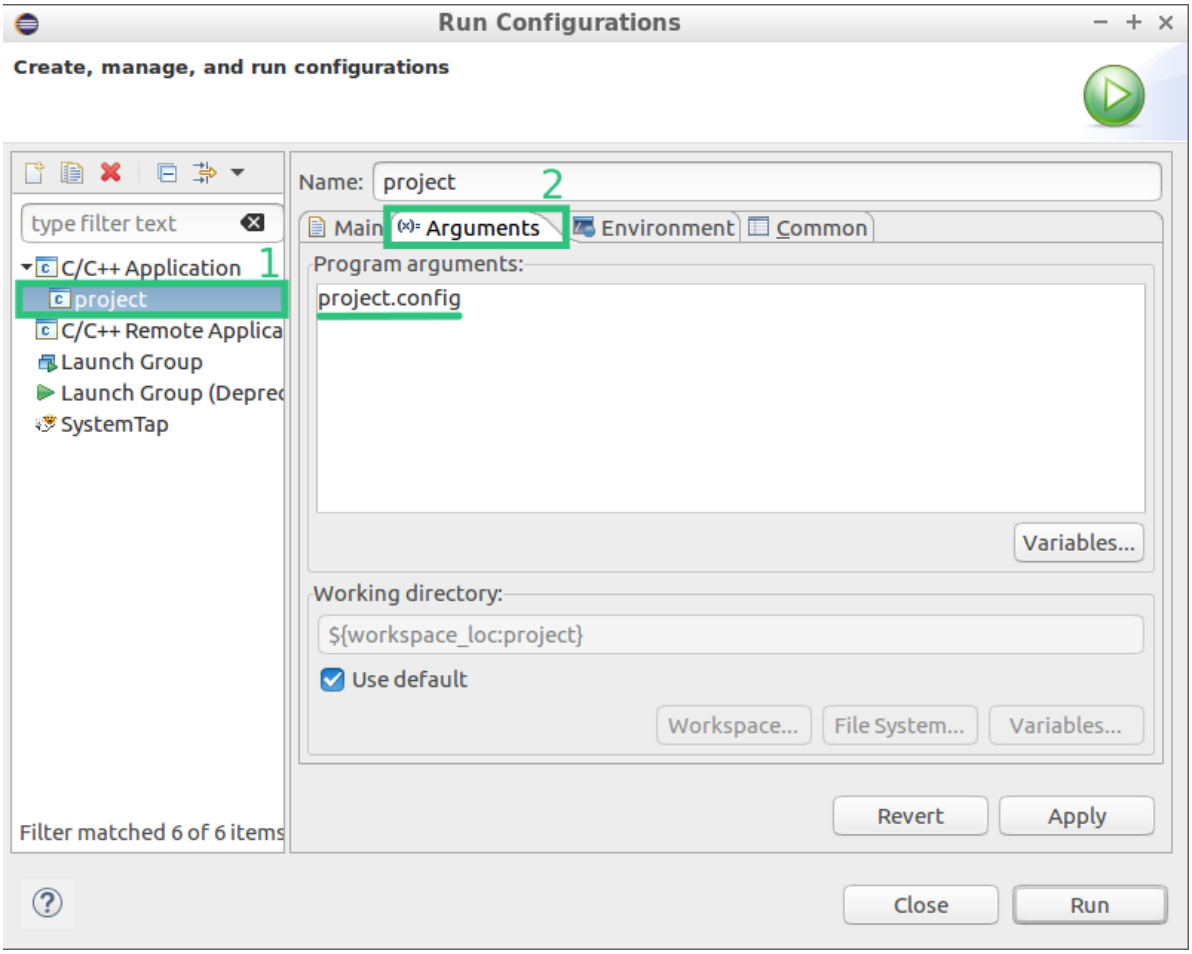
¡Eso es todo! Bueno, casi todo... ¿Cómo podemos lograr esto desde el IDE? 🤔

Pasar argumentos desde Eclipse

Vamos a entrar a las `Run Configurations...` :



Y en la configuración del proyecto nos moveremos a la pestaña `Arguments` , en donde vamos a poner nuestros argumentos separados por espacios:



En caso de usar una **ruta relativa**, es muy importante partir desde el `Working directory` que está configurado ahí más abajo.

Por ejemplo, la variable `${workspace_loc:NOMBRE_DEL_PROYECTO}` apunta hacia la carpeta en donde se encuentra el proyecto Eclipse.

Pasar argumentos desde Visual Studio Code

En el caso de que ya tengamos configurado el debugger, podremos encontrar en nuestro archivo `launch.json` una variable `args` , en donde vamos a poner nuestra lista de argumentos en formato de array de strings.

json

```
{
    "configurations": [
```

```
        {
            // ...
            "args": [ "./una/ruta/a/mi/archivo.cfg" ],
            "cwd": "${workspaceFolder}",
            // ...
        }
    ]
}
```

En caso de usar una **ruta relativa**, es muy importante asegurarnos que la variable `cwd` apunte al valor correcto. En este ejemplo, `${workspaceFolder}` es otra variable que apunta hacia la carpeta que tiene abierta Visual Studio Code[^1].

[^1]: **Visual Studio Code Variables Reference**