A better shell command-line autocompletion for Clang

Yuka Takahashi – Univ. of Tokyo

Google Summer of Code 2017

Mentors: Raphael Isemann and Vassil Vassilev

Normal shell completion (1/2)

- ➢ Collect a list of flags and write a custom shell script.
- ➢ It is a lot of work to collect flags, and portability is low.

Example: `file` command bash autocompletion

```
case $prev in
    --help|-v|--version|-F|--separator) return ;;
    -m|--magic-file|-f|--files-from)
        _filedir return ;;
...
```

Normal shell completion (2/2)

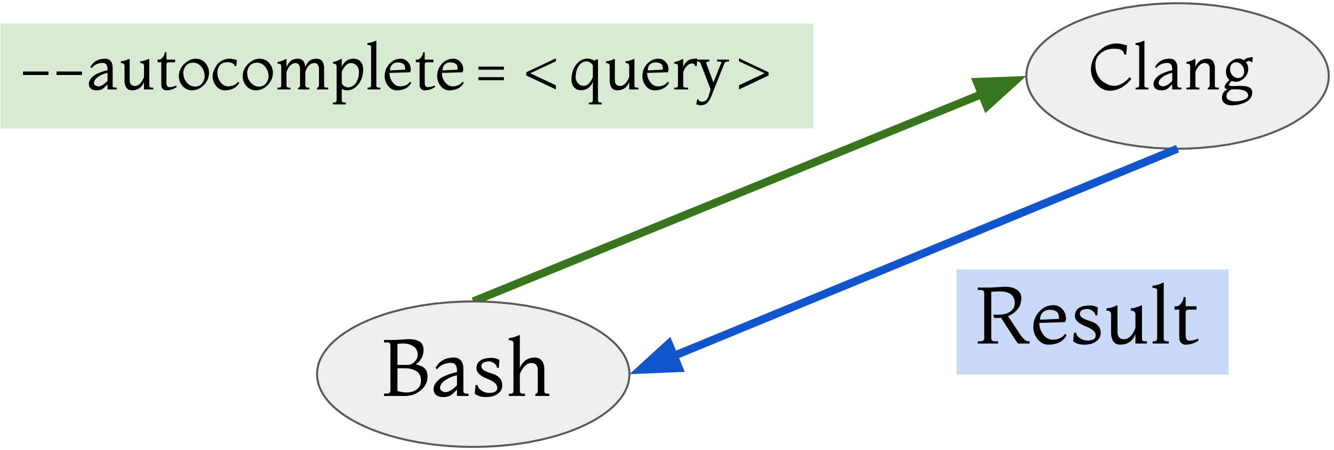
- Script has to parse `help –v`.
- They can’t autocomplete flag’s values because help doesn’t show them.

Example: `gcc` command bash autocompletion

```
COMPREPLY=( $( compgen –W "$( $cc ––help 2> /dev/null | tr '\t' ' ' |\
  command sed –e '/^  *–/!d' –e 's/ *–\([^][ <>]*\).*/–\1/' )" –– "$cur" ) )
[[ $COMPREPLY == *= ]] && compopt –o nospace
```

bash–completion in Clang (1/3)

- We made an API in Clang which dynamically queries available flags.



bash-completion in Clang (2/3)

- You can always complete one flag at a time.
- Just pass this flag to the --autocomplete flag in the selected clang binary.

Example: Clang API for autocompletion

```
$ build/bin/clang --autocomplete=-tr
  -traditional-cpp    Enable some traditional CPP emulation
  -trigraphs    Process trigraph sequences
```

bash-completion in Clang (3/3)

- The API also supports completing the values of flags.
- Shell should provide an incomplete value behind the flag separated by a comma.

Example: Clang API for value autocompletion

```
$ build/bin/clang --autocomplete=-stdlib=,l
  libc++
  libstdc++
```

Shell implementation

- ➢ For `clang –std=[tab]`, shell has to parse the command and execute `clang ––autocomplete=–std=,`

Example: Implementation for bash

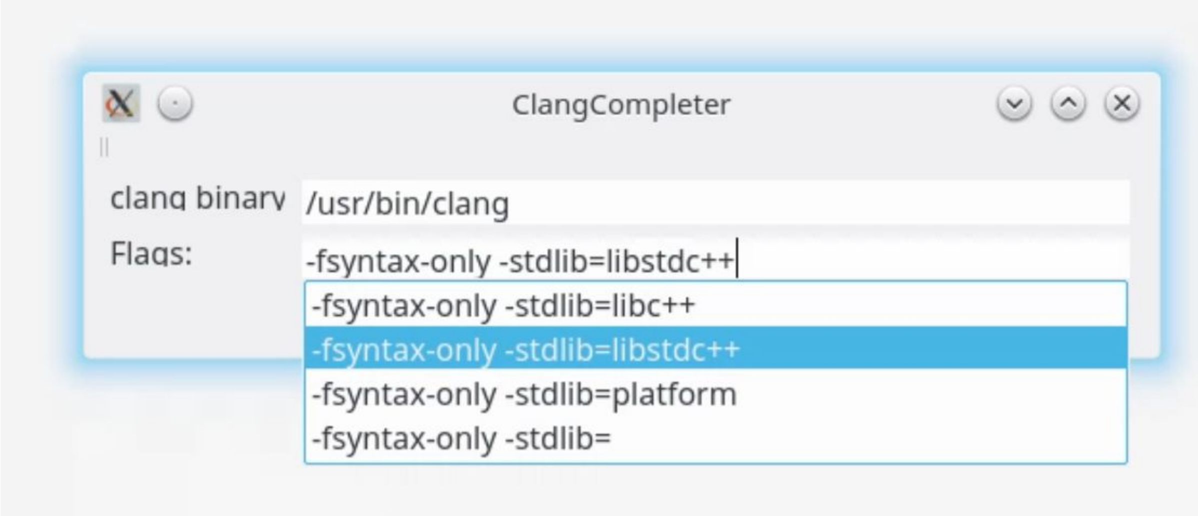
```
if [[ "$cur" = = –* ]]; then
   arg="$arg$cur"
elif [[ "$prev" = = –* && "$cur" = = '=' ]]; then
   arg="$arg$prev=,"
….
flags=$("$path" ––autocomplete="$arg" 2>/dev/null | sed –e 's/\t.*//')
```

IDE implementation

Example: Implementation for Qt

```
QString origArg = arg;
if (arg == "")
   arg = "#";
  if (arg.endsWith("=")) {
   arg += ",";
   completeValue = true;
  } else if (arg.contains("=")) {
....

params <<
  QString("--autocomplete=" + arg);
pingProcess.execute(exec, params);
```



clang binary /usr/bin/clang

Flags: -fsyntax-only -stdlib=libstdc++

-fsyntax-only -stdlib=libc++
-fsyntax-only -stdlib=libstdc++
-fsyntax-only -stdlib=platform
-fsyntax-only -stdlib=

Advantages

- It can support every future version of Clang, starting from Clang 5.0.
- With this API, autocompletion can be built on any shell and has high portability.

Further Information

- LLVM blog:

  blog.llvm.org/2017/09/clang-bash-better-auto-completion-is.html

- GSoC final report:

  summerofcode.withgoogle.com/projects/#6620432915496960

Thanks for your attention!