

補助ツール利用手引き

2014年12月12日(金)

理化学研究所 計算科学研究機構

運用技術部門ソフトウェア技術チーム

寺井 優晃

はじめに

補助ツール: atool (仮名) は、Omni XcalableMPコンパイラのドライバプログラムです。通常のドライバと異なる点は、中間コードを生成した段階で処理を終了させることができます。それ以外は通常のドライバと同様の機能を持っており、例えばプリプロセッサを前処理として呼び出すことも可能です。

現在、atoolはOmni XcalableMPコンパイラの開発元から正式に提供されています。この資料は2013年当時の仕様に基づくもので、一部異なる可能性があることをご留意ください。

ビルド方法

まず、本ツールのソースコードを展開後、トップディレクトリに存在する `configure` を用いてビルド環境を構築します。`configure` の `--prefix` オプションに `Omni XcalableMP` コンパイラのインストールパス(`Omni XcalableMP` 自身の `configure` 発行時に指定した `--prefix` オプションと同じ値) を指定します。

また、デフォルトのネイティブコンパイラとして `gfortran` を使用します。これを置き換える場合、`--with-nativeCompiler` オプションを指定してください。

例えば、`XcalableMP` コンパイラが `/opt/xcalablemp` にインストールされており、ネイティブコンパイラとして `vtfrtpx` を用いる場合:

```
./configure --prefix=/opt/xcalablemp --with-nativeCompiler=vtfrtpx
```

のように `configure` を発行します。さらに、`configure --help` を発行し、その他デフォルト設定の変更が必要となるものについて、必要なオプションを与えることが可能です。

ビルド環境の構築が成功したなら、`make` および `make install` を発行し、本ツールのビルドとインストールを行ってください。

使用方法(1/4)

以下に本ツールの全コマンドラインオプションを示します。

- `--h|--help|--?:`
オプション一覧を表示します。
- `-Wf,opt[,opt...]:`
フロントエンドに直接`opt` をオプションとして渡します。
- `-Wt,opt[,opt...]:`
XcodeML ベース変換器に直接`opt` をオプションとして渡します。
- `-Wd,opt[,opt...]:`
デコンパイラに直接`opt` をオプションとして渡します。
- `-Wn,opt[,opt...]:`
ネイティブコンパイラに直接`opt` をオプションとして渡します。以下に述べる本ツールが解釈してするコマンドラインオプションが、ネイティブコンパイラでも使用される場合に使用します。例えば、`-v` オプションは本ツールも解釈しますし、GCC コンパイラ群も解釈します。ネイティブコンパイラとしてGCC コンパイラを使用しており、`-v` オプションをGCC コンパイラに渡したい場合、`-Wn,-v` のように使用します。

使用方法(2/4)

- -Xf:
フロントエンド段でのXcodeML 出力処理でコンパイル処理を終了します。
- -Xt:
XcodeML ベース変換器出力段でのXcodeML 出力処理でコンパイル処理を終了します。
- -Xd:
デコンパイル段での出力処理でコンパイル処理を終了します。
- -cpp:
強制的にプリプロセッサによる処理を行います。
- -v:
冗長出力を行います。具体的には各処理段でどのようなコマンドが発行されるかを標準エラー出力に出力します。
- -n:
実際のコマンドを発行しません。所謂dryrun モードです。-v オプションと組み合わせることで、実際のコンパイル処理を行わず、その処理で発行されるコマンドシーケンスのみを観察することが可能になります。

使用方法(3/4)

- **-no-clean:**
コンパイル処理の失敗時に、テンポラリファイルや出力途中となっているターゲットファイル等の削除を行いません。
- **-ignore-output:**
-o コマンドで指定されたターゲットファイル名を使用せず、デフォルトの出力ファイル名を使用します。例えば、元々のビルドシーケンス上 `gfortran -c foo.F90 -o foo.o` のコマンドで処理された `foo.o` がターゲットになる時、この処理をフロントエンド段で中止するために Makefile 内の `FFLAGS` に単純に `-Xf` を追加するだけだと `foo.o` に `XcodeML` が出力されてしまって不都合がある場合にこのオプションを指定すれば、`foo.o` の代わりに `foo.xml` が出力ファイルとなります。
- **-emit-cpp-result:**
プリプロセッサ段の出力を `.i` を拡張子とするファイルに出力します。コンパイル処理自身は指定された処理段まで継続します。
- **-emit-frontend-result:**
フロントエンド段の出力を `.xml` を拡張子とするファイルに出力します。コンパイル処理自身は指定された処理段まで継続します。
- **-emit-translator-result:**
フロントエンド段の出力を `.trans.xml` を拡張子とするファイルに出力します。コンパイル処理自身は指定された処理段まで継続します。

使用方法(4/4)

- `-emit-decompile-result:`
デコンパイル段の出力を`.decomp.f90` を拡張子とするファイルに出力します。コンパイル処理自身は指定された処理段まで継続します。
- `--cpp-command=cmd:`
プリプロセッサとして`cmd` を使用します。
- `--translator-command=cmd:`
XcodeML ベース変換器として`cmd` を使用します。
- `--native-compiler-command=cmd:`
ネイティブコンパイラとして`cmd` を使用します。 `--cpp-command`によるプリプロセッサの指定がなされていない場合、`cmd` がプリプロセッサとしても使用されます。

上記に該当しないオプションはすべて、必要に応じて、ネイティブコンパイラに渡されません。

補助ツールとしての利用

本ツールをK-scopeの補助ツールとして使用する場合、既存のビルド環境に於けるFortran コンパイラを本ツールとして設定し、さらに本ツールに`-emit-frontend-result` オプションを渡すようMakefile 等を変更することが最も簡単な方法となります。これにより、通常のビルドターゲットを変更することなく、コンパイル対象となるソースファイル全てについて、`.xml` を拡張子とするファイルにフロントエンド処理出力が生成されます。

使用例

```
atool -O0 -Xf -I/home/teraim/opt/mpich/include -cpp -emit-frontend-result -Wf,-endlineno
```

使用例(京のフロントエンド)

```
atool -O0 -Xf -I/home/teraim/opt/mpich/include -cpp --cpp-command=frtpx -emit-frontend-result -Wf,-endlineno
```