

## UTSの言語プロセッサとプログラムライブラリ

二村, 祥一  
九州大学大型計算機センター

<https://doi.org/10.15017/1468157>

---

出版情報：九州大学大型計算機センター広報. 20 (5), pp.441-449, 1987-09-25. 九州大学大型計算機センター  
バージョン：  
権利関係：

## UTSの言語プロセッサとプログラムライブラリ

二村 祥一\*

## 1. 利用可能な言語プロセッサ

現在, UTSのもとで使用可能な言語プロセッサとしては次のものがある.

・as	UTSアセンブラ[1]
・cc	Cコンパイラ[1]
・fcc	富士通Cコンパイラ[2]
・f77	Fortran77コンパイラ[1]
・ratfor	Ratforトランスレータ[1]
・efl	拡張Fortranコンパイラ[1]
・pc	Pascalコンパイラ[1]
・sno	Snobolコンパイラ, インタプリタ[1]
・prolog	Prologインタプリタ[3]
・yacc	コンパイラ・コンパイラ[4]
・lex	字句解析プログラム生成[4]

UTSには, ccとfccの二つのCコンパイラがある. ccはアムダール社製のCコンパイラであり, fccは富士通社製のCコンパイラである. ratfor, eflは, いずれもFortranのプリプロセッサである. Ratfor(Rational Fortran)やEFL(Extended Fortran Language)のプログラムは, Fortranのプログラムに比べて見やすい. EFLはC言語のような制御構造, データ構造をもっている. prologは, 富士通社製のPrologインタプリタであり, その処理系の仕様は基本的にはDEC10 Prologに準拠している.

yacc(yet another compiler-compiler)は, インタプリタやコンパイラのゼネレータであり, 結果としてC言語のプログラムを出力する. Ratfor, Basicなど多くの言語プロセッサがyaccを用いて作成された. lex(lexical analyzer)は, 字句解析を行うC言語プログラムを生成するツールである.

なお, 富士通社製のFortran77コンパイラが, 近いうちに提供される予定である.

ここでは, CコンパイラおよびFortran77コンパイラの使用法について述べる. UTSのFortran77は, MSPのFortran77とは利用環境が大きく異なる. f77コマンドの利用法はccコマンドのそれに近い.

---

昭和62年8月3日受理

\* 九州大学大型計算機センター

## 2. Cコンパイラの利用法

### 2.1 UTS-Cの特徴

九大センターでは、本年6月にOS IV/F4 MSPのもとで動作するC言語を公開した[11].

C言語は、もともとUNIXオペレーティングシステムの開発言語として設計された。それは高級言語としての性格も併せもち、UNIXのソフトウェア全体にわたっての記述言語として利用されてきた。そして、UNIXの成長とともに、Cのプログラム開発環境も整備されていった。UTSは、UNIX System Vを基本にして、パークレイ版UNIXの機能及び富士通独自の機能を追加したものであり、C言語に関して、MSPとは異なった快適なプログラム開発環境を与える。

C言語のプログラムの翻訳から結合編集までを行うコマンドとして、ccやfccがあり、それらはオプションの指定に従って各種の機能を果たすように設計されている。Cの実行形式プログラムは、コマンドインタプリタであるシェルのもとで動作する。そこでは、シェルのもつ入出力リダイレクション機能やパイプ機能などが使用できる[8].

また、C言語のプログラム開発環境として、次のようなツールが用意されている。

make, cb, lint, prof, cflow, sdb, ctrace

makeは、ファイル作成の時間関係を利用して、翻訳処理を効率的に管理するツールである。cb(C-Beautifier)はソースプログラムの美化(清書)ツールであり、プログラムを見やすくする。lintは、Cソースプログラムを翻訳することなく、文法上の誤りなどを検出するツールである。profはプログラム改善のために、実行プロファイルをとるためのツールである。cflowはCプログラムの関数呼出し関係を解析し、コールグラフを作成するツールである。sdb, ctraceはプログラムのデバッグのためのツールである。

Cの言語仕様については、参考文献[1, 2, 10]を参照されたい。

### 2.2 Cソースプログラムの編集とラインプリンタ出力

Cソースプログラムは、UNIXの標準画面エディタであるviや富士通社製の画面エディタnedなどを用いて作成できる[7].

日本語文字を含むCソースプログラムの作成には、jviが使用できる。この場合、端末を日本語モードにするために、次のようなコマンドを入力しておく必要がある。

```
jstty -c sjis -t -J
```

(又は簡単に、jstty -s-t でよい)

このコマンドは、端末のコード系をシフトJISに、端末モードを日本語モードに、更に端末の状態をjviモードにすることを指示する。

これらのエディタのほかに、C言語の構造エディタであるscope(Structured C-Oriented Program Editor)を使用することもできる。

Cソースプログラムのラインプリンタ出力については、UTSにはラインプリンタがないため、一旦UTSのファイルをMSPにファイル転送しなければならない[9]。このためには `utocp` コマンドを用いる。ファイル転送を行うと同時にラインプリンタに出力することもできる。この場合は、次のようなジョブをファイル `.utosendf` に用意すればよい。

```
//a79999aa job password,class=a
// exec pgm=utorcv
//outfile dd dsn=~TOFILE,disp=shr
//reader dd .
~FROMFILE
// exec pscopy
//sysut1 dd dsn=~TOFILE,disp=shr
//sysut2 dd sysout=a,ucs=as10
//
```

### 2.3 プログラムの翻訳と実行

Cソースプログラムの翻訳から結合編集までを行うためのコマンドとして、`cc`と`fcc`がある。`cc`はアムダール社製のCコンパイラによる翻訳を、また`fcc`は富士通社製のCコンパイラによる翻訳を行うためのものである。`cc`と`fcc`は機能にそれほど大差はない。ここでは、`cc`コマンドを使って説明する。

`cc`コマンドの形式は次のとおりである。

```
cc ファイル名... [オプション]...
```

`cc`は以下の処理単位からなる。

- ・プリプロセッシング (cppによる)
- ・翻訳 (ccomによる)
- ・最適化 (c2による)
- ・アセンブル (asによる)
- ・リンク (ldによる)

ファイル名が `.c` で終わるファイルはC言語ソースプログラム、またファイル名が `.s` で終わるファイルはアセンブラソースプログラムとして扱われる。各ソースプログラムからオブジェクトプログラムが作成される。入力ファイルの内容識別修飾子を `.o` に変えたファイルが現行ディレクトリに作成され、結果のオブジェクトプログラムが入れられる。すべてのオブジェクトプログラムが、リンク `ld` に入力され、一つの実行可能なオブジェクトプログラムになる。このファイル名は、標準的には `a.out` である。翻訳の過程で作成されるオブジェクトプログラムのファイルが一つの場合、それ (`.o` のファイル) は削除される。オブジェクトプログラムのファイルが複数の場合、それらは保存される。

次にccコマンドのオプションをまとめる。

- c 入力されたソースプログラムに対してアセンブルまでの処理を行う。結果は、入力ファイルの内容識別修飾子を.oに変えたファイルに作成する。
- p prof(実行プロファイル)用にオブジェクトファイルを作成する。
- g sdbの使用に必要な追加情報を出力するようコンパイラに指示する。
- o ファイル名  
a.outに代わるオブジェクトプログラムのファイル名を指定する。
- O オブジェクトコード・オブティマイザを呼び出す。
- S Cのソースプログラムを翻訳し.sで終わるファイル名をもつアセンブリ言語の出力を生成する。
- E Cのソースプログラムに対してcppだけを実行し、その結果を標準出力に送る。
- P Cのソースプログラムに対してcppだけを実行し、その結果を.iで終わるファイルに書き出す。
- Wx,引数1[,引数2...]  
引数iをxで示された処理単位に渡す。xには[po2al]のうちの一つが指定できる。それぞれプリプロセッシング、翻訳、最適化、アセンブル、リンクに対応する。

使用例1) cc prog.c

<説明> ソースプログラムprog.cを翻訳・結合編集する。実行形式のオブジェクトモジュールがa.outに作られる。cshの促進記号%に対して、a.outを入力するとプログラムが実行できる。

使用例2) cc xyz.c -o pqr

<説明> ソースプログラムxyz.cを翻訳・結合編集する。実行形式のオブジェクトモジュールがファイルpqrに作られる。cshの促進記号%に対して、pqrを入力するとプログラムが実行できる。

使用例3) cc sqrt.c -Wl,-lm

<説明> ソースプログラムsqrt.cを翻訳・結合編集する。リンクldにオプション-lmを渡している。-lmはFortranの関数呼出しを行う場合に指定する。

使用例4) cc x.c y.c z.c

<説明> ソースプログラムx.c, y.c, z.cを翻訳・結合編集する。オブジェクトモジュールのファイルx.o, y.o, z.oが作られ保存される。実行形式のオブジェクトモジュールはファイルa.outに作られる。

### 3. Fortran77コンパイラの利用法

#### 3.1 UTS-Fortran77の特徴

ここでは、UTSのFortran77コンパイラの特徴のうち、MSPのFortran77の利用者にとって特筆すべきものをいくつか紹介する。

- ・ Fortranソースプログラムを英小文字で記述できる。英大文字と英小文字を区別するかどうかはオプションで指定する。
- ・ Fortranの入出力機番5, 6, 0は、それぞれ標準入力、標準出力、標準エラー出力に結合する。それ以外の入出力機番nはファイル名がfort.nのファイルに結合する。
- ・ 関数の再帰呼出しが行える。
- ・ 2進定数が扱える。
- ・ RatforやEFLを、Fortranのプリプロセッサとして利用できる。
- ・ プログラムのデバッグのためにsdb(記号デバッガ)が使用できる。
- ・ プログラムの改善のためにprof(実行プロファイル)が使用できる。
- ・ UTSのコード系はASCIIであり、MSPのコード系はEBCDICである。両システム間でプログラムを移植する場合、コード系に依存する部分については書き替えが必要である。

### 3.2 Fortranソースプログラムの編集とラインプリンタ出力

Fortranソースプログラムの編集には、画面エディタvi, nedなどが使用できる[7]。また、ソースプログラムのラインプリンタ出力には、ファイル転送コマンドutocpを使う。詳しくは、2.2節を参照されたい。

### 3.3 Fortranプログラムの翻訳と実行

Fortran77ソースプログラムの翻訳から結合編集までを行うコマンドとしてf77がある。f77コマンドの形式は次のとおりである。

```
f77 [オプション]... ファイル名...
```

このコマンドは以下の処理単位からなる。

- ・ プリプロセッシング(eflによる)
- ・ プリプロセッシング(ratforによる)
- ・ 翻訳(f77による)
- ・ 翻訳(cgenによる)
- ・ 最適化(c2による)
- ・ アセンブル(asによる)
- ・ リンク(ldによる)

ファイル名が.fで終わるファイルは、Fortran77ソースプログラムとみなされて翻訳される。各ソースプログラムからオブジェクトプログラムが作成される。入力ファイルの内容識別修飾子を.oに変えたファイルが現行ディレクトリに作成され、結果のオブジェクトプログラムが入れられる。

.r又は.eで終わる名前をもつファイルは、それぞれRatforソースプログラム、又はEFLソースプログラムとみなされる。これらは、まず対応するプリプロセッサによりFortranソースプログラムに変換される。続けて、それらはf77により翻訳され.oのファイルが作成される。同様に、.c又は.sで終わる名前をもつファイルは、C言語又はアセンブリ言語のソースプログラムとみなされて翻訳される。

すべてのオブジェクトプログラムが、リンカldに入力され、一つの実行可能なオブジェクトプログラムが作られる。そのファイル名は、標準的にはa.outである。

個々の処理単位は、オプションによって迂回させることができる。

次にf77コマンドのオプションをまとめる。

- c 入力されたソースプログラムに対してアセンブルまでの処理を行う。結果は、入力ファイルの内容識別修飾子を.oに変えたファイルに作成する。
- p prof(実行プロファイル)用にオブジェクトファイルを作成する。
- O オブジェクトコード・オブティマイザを呼び出す。
- S 指定プログラムを翻訳し.sで終わるファイルにアセンブリ言語の出力を行う。
- o ファイル名  
a.outに代わるオブジェクトプログラムのファイル名を指定する。
- g sdbの使用に必要な追加情報を出力するようコンパイラに指示する。
- 66 66Fortranとの互換を保つ拡張を抑制する。
- C 実行時の添字範囲検査用のコードを生成する。
- U Fortran77ソースプログラムで大文字と小文字を区別して扱う。  
(通常、f77では大文字と小文字は区別しない)
- v 翻訳中の各プロセスごとの診断メッセージを出力する。
- w 全ての警告メッセージを抑制する。
- F 関連ファイルにelf, ratforプリプロセッサを適用し、その結果を.fのファイルに出力する。
- m elf, ratforプリプロセッサで変換する前に、それらのソースファイルに対してm4プリプロセッサを適用する。

#### 使用例1) f77 main.f

<説明> ソースプログラムmain.fを翻訳・結合編集する。実行形式のオブジェクトプログラムがa.outに作られる。cshの促進記号%に対して、a.outを入力するとプログラムが実行できる。

#### 使用例2) f77 -o test xyz.f

<説明> ソースプログラムxyz.fを翻訳・結合編集する。実行形式のオブジェクトプログラムがファイルtestに作られる。cshの促進記号%に対して、testを入力することによりプログラムが実行できる。

#### 使用例3) f77 main.f sub1.f sub2.f

<説明> ソースプログラムmain.f, sub1.f, sub2.fを翻訳・結合編集する。オブジェクトプログラムがファイルmain.o, sub1.o, sub2.oに作られる。実行形式のオブジェクトプログラムはa.outに作られる。

#### 4. プログラムライブラリ

C及びFortran77のためのプログラムライブラリは、マニュアルUTSプログラミング文法書[5]では次のように分類されている。

分類	ライブラリの内容
2	システムコール・ライブラリ(libc)
3C	C及びアセンブラライブラリ(libc)
3S	標準入出力ライブラリ(libc)
3M	数学ライブラリ(libm)
3F	Fortranライブラリ(libF77)
3X	その他

ccコマンドは、標準Cライブラリlibc(2, 3C, 3S)を自動的にロードする。ただし、Cソースプログラム中に該当するインクルードファイルを指定しておく必要がある。例えば、標準入出力ライブラリを組み込むには、次のインクルード文を指定する。

```
#include <stdio.h>
```

ccでは、数学ライブラリ(3M)を組み込むこともできる。数学ライブラリを組み込むには、次のインクルード文を指定する。

```
#include <math.h>
```

この場合は、更にリンクldに渡すオプションとして-lmを指定する。これは、ccコマンドでは次のように記述する。

```
cc test.c -Wl,-lm
```

f77コマンドは、Fortranライブラリ(3F)、数学ライブラリ(3M)を自動的に組み込む。UTSでは、現在、富士通社製の科学計算用サブルーチンライブラリSSL IIが使えないが、来年はじめには使用可能になる予定である。

#### 5. C及びFortran77プログラムの翻訳・結合編集・実行例

C及びFortran77プログラムを翻訳・結合編集を行い、実行するまでの例を示す。

下線部は、利用者の入力部分である。

##### 5.1 Cプログラムの翻訳・結合編集・実行

次のプログラムは、平方根を求めて表示するためのものである。

```
% cat sqrt.c
#include <math.h>
main(){
    int i;
    double x,y,sqrt();
    for(i=1; i<13; i++) {
        x=i;
        y=sqrt(x);
        printf(" %d %f\n",i,y);
    }
}
```

: プログラムsqrt.cの表示.

```
% cc sqrt.c -o aaaa -Wl,-lm
```

:sqrt.cの翻訳と結合編集. オブジェクトモジュールはファイルaaaaに作る. 数学ライブラリsqrtの組み込みを指示している.

```
% aaaa
```

:aaaaの実行.

```
1 1.000000
2 1.414214
3 1.732051
4 2.000000
5 2.236068
6 2.449490
7 2.645751
8 2.828427
9 3.000000
10 3.162278
11 3.316625
12 3.464102
%
```

## 5.2 Fortran77プログラムの翻訳・結合編集・実行

次のプログラムは, 英大文字を英小文字に変換するためのものである.

```
% cat exam.f
character a*20
a='CAPTAIN FUTURE NO.1'
write(*,*) a
call cnvlc(a)
write(*,*) a
end
```

: プログラムexam.fの表示.

```

subroutine cnvlc(str)
character str*(*)
do 10 i=1,len(str)
    if(str(i:i).ge.'A'.and.str(i:i).le.'Z')
+       str(i:i)=char(ichar(str(i:i))+32)
10 continue
end

```

% f77 -o lower exam.f :exam.fの翻訳と結合編集. オブジェクトモジュールはファイルlowerに作る.

% lower :lowerの実行.

CAPTAIN FUTURE NO.1

captain future no.1

%

#### 参考文献

1. 計算機マニュアル(英語版), UTS Programming Guide(24SP1061E-1), 富士通(株).
2. 計算機マニュアル, UTS C言語手引書(24SP-3060-1), 富士通(株).
3. 計算機マニュアル, FACOM PROLOG手引書(99SP-5760-1), 富士通(株).
4. 計算機マニュアル(英語版), UTS Support Tools Guide(24SP1081E-1), 富士通(株).
5. 計算機マニュアル, UTS プログラミング文法書(24SP-1031), 富士通(株).  
(英語版): UTS Programmer Reference Manual(24SP1031E-1), 富士通(株).
6. 計算機マニュアル, UTS コマンド文法書(一般利用編)(24SP-1021), 富士通(株).  
(英語版): UTS User Reference Manual(24SP1021E-1), 富士通(株).
7. 松尾文碩: UTSのエディタ, 九州大学大型計算機センター広報, Vol.20, No.5, 1987.
8. 松延栄治: UTSのシェルとファイルシステム, 九州大学大型計算機センター広報, Vol.20, No.5, 1987.
9. 二村祥一: UTS-MSP間のファイル転送, 九州大学大型計算機センター広報, Vol.20, No.5, 1987.
10. B.M.Kernighan and D.M.Ritchie : The C programming language, Prentice-Hall, 1978.  
(石田晴久訳: プログラム言語C UNIX流プログラム書法と作法, 共立出版, 1981.)
11. C言語の公開について, 九州大学大型計算機センターニュース, No.353, 1987.