

C-FORTRAN 혼합언어 (Mixed Language) 프로그래밍

최 창 근*

1. 서론

최근 computer 관련기술의 급속한 발달에 따라 PC의 공학적 이용이 급속히 확대되고 있으나 하드웨어(H/W)의 발전을 소프트웨어(S/W)의 개발이 미처 뒤따르지 못하고 있는 실정이다. 현대 산업계의 programming language에 대한 요구도 점점 다양화 되어 새로운 종류의 programming language가 탄생하기도 하고, 기존의 언어들도 보완을 거듭하면서 이러한 요구를 충족시켜나가고 있는 추세이다.

전에는 범용언어(General Purpose Language) 만으로도 소기의 목적을 달성할 수 있었으나 산업이 점차 고도화되고 복잡해짐에 따라 특정분야에서의 능력이 탁월한 objective language(예를 들면 LISP, SMALL TALK, PROLOG 등)에 의한 문제해결도 시도되고 있다. 특히 용량이 제한된 PC에서는 각각의 module을 그 module에 가장 유리한 언어로 만들어 각기 다른 언어로 만들어진 module들을 서로 연결하여 program을 완성하는 소위 혼합언어 프로그래밍 기법이 요구되기도 한다. 예를 들어 C와 Assembly어를 연결하면 C가 하기 힘든 기계자체와의 "대화"를 통한 신속한 프로그램 수행을 Assembly어가 할 수 있고 C는 범용 언어로서의 장점인 code화의 용이성, I/O의 용이성을 제공하게되어 상호보완적으로 프로그래밍 할 수 있다. 그리고, LISP와

C를 연결하면, 인공지능의 활용에 유리한 LISP는 사고와 판단을, C는 graphics 기능을 담당하는 프로그래밍이 가능하다는 점은 이미 널리 알려진 바이다.

이처럼 두 가지 또는 그 이상의 언어를 각각의 언어의 장점만을 취하여 혼합한다면 매우 효율적인 프로그램을 만들 수 있게 된다. 이 글에서는 공학문제 해결에 많이 쓰여온 FORTRAN과 C언어의 PC에서의 Interface에 대해 논하고 구조공학 분야에서의 활용방안의 일단을 제시하고자 한다.

2. C-FORTRAN의 관계

FORTRAN은 이공학을 전공하는 사람들이 오랫동안 사용해 온, 과학계산에 우수한 언어이다. Double precision이나 복소수의 처리등, 수치계산능력에서는 타 언어에서는 찾아보기 힘든 장점이 있다. FORTRAN 77이 소개된 이후 LOOP를 보완하게 되어 structured programming에 대한 문제도 많이 개선된 바 있으나 아직 Graphics 기능이 기본 option이 아니라는 단점이 있다. 이에 비해 C 언어는 graphics에 있어서 타 언어와 비교되지 않을 만큼 융통성(flexibility)을 갖고 있다.⁴⁾ 따라서 FORTRAN으로 수치계산(구조계산)을 수행하고 나서 그 결과를 C 언어로 출력(graphics 포함)하는 것이 C-FORTRAN 혼합언어 프로그래밍의 좋은 활용 예가 될 것이다. 이 경우 지금까지 개발된 많은 FORTRAN 프로그램의 개선을 위해서 프로그램 전체를 C로 변

* 정회원, 회장, 한국과학기술원 교수, 공학박사

환할 필요 없이 추가적인 기능을 C에 의한 프로그래밍으로 해결할 수 있다. 우선 실용적인 프로그래밍 방법2가지를 들어본다면,

(1) C로 직선과 곡선(원)등을 그리고, 원하는 위치에 원하는 문자나 부호를 삽입하거나, 주어진 함수를 주어진 구간에 걸쳐 plot하는 function을 만들어서 FORTRAN에서 그것들을 필요할 시기에 CALL을 통해 사용하는 방법이 있다. 이 방법은 C 쪽의 code가 간단해지는 장점이 있으며 FORTRAN programmer가 마치 graphics 기능이 있는 언어로 프로그래밍 하는 것처럼 쉽게 plot할 수 있는 장점이 있다. 이 방법은 Grafmatic⁷의 개발자들이 쓴 방법이다.

(2) 일단 FORTRAN에서 모든 수치 계산을 끝낸 뒤 그 결과를 COMMON Block이나 보조 기억장치 등에 저장하고, C에서 이 FORTRAN의 DATA를 불러서 plot하는 방법이 있다. 이것은 menu-driven 형식으로 사용자와 interactive하게 작업을 진행하면서 video mode를 자주 바꾸어야 할 경우나, FORTRAN에 의한 계산의 중간 중간에 plot를 통한 확인이 필요 없을 경우에 효율적이다.

또한 C와 FORTRAN 사이의 Data의 교환(convey)에는 다음과 같은 몇가지 유의할 점이 있다.

첫째, C와 FORTRAN의 기본적인 변수의 크기(variable size)가 다르다는 점이다. 예를 들면 C는 정수인 경우 2 byte를 쓰는데 비해 FORTRAN은 4 byte를 쓴다. 그렇기 때문에 변수를 지정할 때 그 크기를 어느 한 쪽으로 통일해야 한다는 것이다.(자세한 것은 표1에 정리하였다.)

둘째, FORTRAN의 COMPLEX라는 개념이 C 언어에는 없다는 점이다. 복소수를 C에서 쓰기 위해서는 2차원 array를 만들던가, struct 문을 써서 2개의 unit로 나누어야 한다.

셋째, C에는 FORTRAN의 COMMON이 없다. 그러므로 FORTRAN의 COMMON에 대응하는 struct 문을 만들어 FORTRAN의 data를 불러와야 한다.

넷째, Microsoft C(5.0)의 경우, compile 시의 default memory model이 SMALL 인데 비해 Microsoft FORTRAN (4.01)의 default memory

model은 LARGE이므로, C의 compile 시 LARGE model로 compile 되도록 조건을 달아주어야 한다.(Microsoft의 경우 /AL)

표1. C와 FORTRAN의 비교

FORTRAN TYPE	C TYPE
INTEGER*2	int *
CARG(INTEGER*2)	int
INETEGER*4	long int *
CARG(INTEGER*4)	long int
REAL*4	flot *
CARG(REAL*4)	float(double pushed on stack)
REAL*8(double precision)	double *
CARG(REAL*8)	double
COMPLEX	struct complex
COMPLEX	float[2] (실수부가 첫째 array에)
COMPLEX*16	struct
COMPLEX*16	double[2](실수부가 첫째 array에)
LOGICAL*1	char *(Boolean Value)
LOGICAL*4	-
CHARACTER	char[]
CARG(CHARACTER)	char[]
label	-
EXTERNAL	-
CHARACTER*(*)	struct CHARACTER { char *text; int length; }

3. 프로그래밍 예

이상의 기초적인 지식하에 제(2)의 방법에 의한 Microsoft C와 Microsoft FORTRAN의 혼합 프로그래밍을 예시하고자 한다.

FORTRAN과 연결하기 위해 우선 C program의 맨 앞에

```
void extern fortran builds( ) ;
```

라고 선언하고 다음과 같이 struct문을 만들어야 한다.

```
struct bb{
```

```
float sd[19] ;
float xdis[10] ;
float PBX[4] [6] [19]
```

```
extern struct bb fortran B ;
```

여기서 fortran이란 Microsoft 사의 compiler들 사이에서만 통하는 것으로 C program에 FORTRAN 과 interface하게 되므로 그 쪽에 맞추어서 변수 들을 집합시키라는 뜻이 된다.⁴⁾ 그러므로 대응하는 FORTRAN의 변수에 대해서는 case sensitive 하지도 않고 심지어 같은 block-struct에서는 순서만 같으면 변수나 array를 넘겨 받게 된다.

위의 struct bb에 대응하는 FORTRAN의 source code는 다음과 같다.

```
subroutine builds
INTEGER*2 NST, NCP
REAL SD, PD
COMMON /NUM/ NST, NCP
COMMON /B/ SD(19), PD(10)
COMMON /P/ PBX(19, 6, 4)
```

```
..
(FORTRAN PROGRAM)
```

(여기서 FORTRAN의 real이 C에서는 float가 된 것에 주의)

이 단계에 이르기까지 C에서는 FORTRAN의 어느 data를 사용하겠다고 선언했고 FORTRAN에서는 그것을 define한 셈이다.

따라서 다음 단계로 C에서 실제로 그 data들을 불러서 활용하는 단계를 보기로 한다.

```
for(cc=0;cc<19;cc++)
{
  for(rb=0;rb<6;rb++)
  {
    for(ra=0,txl=0;ra<NUM,NFX;ra++)
    {
      txl+=B.xdis[cc];
```

```
if(max<P.PBX[ra] [rb] [cc])
max=P.PBX[ra] [rb] [cc];
```

```
..
(C PROGRAM)
```

```
..
```

여기서 NFX는 FORTRAN의 NUM이란 block에 있고(값은 4) xdis()는 FORTRAN의 B block에 PD라는 변수로 정의되어 있다. 이처럼 FORTRAN과 같은 이름 NFX를 써도 되고 xdis라는 새로운 이름으로 PD를 불러도 된다. 여기서 array에 대해 조심할 것은 FORTRAN은 array를 column-major로 다루는데 비해 C는 row-major로 다룬다는 사실과 FORTRAN의 array subscript는 1, 2, 3, 4, …으로 나가는데 비해 C에서는 0, 1, 2, 3, …으로 나간다는 점이다. 다시 말하면 만약 FORTRAN에서 array가 F(i, j, k) 였다면 C에서는 F[k-1] [j-1] [i-1]이 되어야 한다는 것이다.⁵⁾

이상의 routine이 포함된 FORTRAN과 C의 source file의 이름들을 각각 builds. for, plot. c라 한다면, FORTRAN에서는

```
FL /C BUILDS. FOR
```

```
로 compile 해야 되고 C에서는
CL /C /AL PLOT. C
```

로 compile 해야 한다. FORTRAN의 compiler와 source code가 있는 Directory를 C:\MFOR4 라 하자. 이 경우에는 C에서 FORTRAN을 불렀으므로 C 쪽에서 link 해야한다. C의 compiler와 source가 있는 directory에서

```
LINK PLOT. OBJ C:\MFOR4\BUILDS. OBJ:
로 LINK하면 C가 있는 directory에 plot. exe file이 생긴다.
```

PC에서의 C-FORTRAN 혼합언어 프로그래

밍이 현단계로는 아직 초보적인 단계로서 많은 제한이 있는 것도 사실이다. Microsoft C 에 있어서는 문자의 크기와 방향(orientation)을 현재로는 자유자재로 바꿀 수 없기 때문에 사용자의 필요에 따라 사용자 자신이 font를 만들어 크기나 방향을 조정해야 한다.³⁾ TURBO C(Borland Inc.)를 혼합언어 프로그래밍에 사용할 때는 interface가 가능한 FORTRAN compiler는 F77L(Lahey Inc.) 뿐인데, 지금(이 원고를 쓰는 시점)은 TURBO C 1.5까지만 interface가 가능하고 TURBO C 2.0까지 가능하게 되려면 F77L version이 upgrade 되어야 한다.⁶⁾

최근에는 혼합언어 프로그래밍 이외에도 단일언어 프로그래밍에서 해당언어의 기능 확장에 의해 해당 언어의 단점을 보완하여 혼합언어 프로그래밍의 필요성을 축소해 나가는 경향이 있다. 즉, FORTRAN의 graphics 기능을 보완하면 굳이 다른 언어(C)와의 혼합 프로그래밍이 아니더라도 사용자의 필요를 어느 정도 충족시켜줄 수 있을 것이다. 특히 기존 프로그램의 기능 강화와 확장에 그러하다. 실제로 FORTRAN의 graphics 기능 강화를 위한 상업적 소프트웨어(S/W)로 Grafmatic, Printmatic, Plotmatic⁷⁾ 등이 나와있다.

4. 결론

PC에서의 C와 FORTRAN의 혼합언어 프로그래

밍은 각 언어의 장점을 최대한 활용하고 단점을 상호보완하게 되어 이상적이라 할 수 있다. 그러나 완벽한 혼합언어 프로그래밍은 소프트웨어(S/W)의 추가개발, 사용자의 두 언어에 대한 지식의 필요 등이 요구된다. 또한 단일언어 프로그래밍에서도 그 언어의 단점 보완등으로 혼합언어 프로그래밍과 어느 정도 같은 효과를 거둘 수도 있는 면도 있다. 어느 방법을 사용하느냐하는 문제는 사용자의 기호와 능력, 그리고 사용 가능한 소프트웨어에 달려다고 하겠다.

참 고 문 헌

1. Microsoft C 5.0 Optimizing Compiler Mixed Language Programming Guide Microsoft Inc. 1987.
2. Herbert Schildt, C The Complete Reference Osborne / McGraw-Hill, Berkeley 1987.
3. Robert Lafore, The Waite Group's Microsoft C-Programming for the PC Howard W. Sams & Company Indianapolis, 1989.
4. Nelson Johnson, Advanced Graphics in C Osborne / McGraw-Hill, Berkeley 1987.
5. Micheal A. Floyd "Making the C-to-FORTRAN Connection" Dr. Dobb's Journal, Annual C Issue, #154, August, 1989.
6. Turbo C 2.0 User's Manual Borland Inc. Scotts Valley, 1988.
7. Grafmatic, Printmatic, Plotmatic User's Manual Microcompatibles Inc. Silver Spring 1987.