

---

Subject: stupid problem in fortran  
Posted by [gino](#) on Mon, 22 Apr 1985 00:49:46 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Article-I.D.: sdchema.370  
Posted: Sun Apr 21 19:49:46 1985  
Date-Received: Tue, 23-Apr-85 06:48:25 EST  
Reply-To: [gino@sdchema.UUCP](mailto:gino@sdchema.UUCP) (Eugene G. Youngerman)  
Organization: U.C. San Diego Chemistry Dept  
Lines: 26  
Xref: watmath net.micro.pc:3745 net.lang.f77:267

This is really dumb.....

I have been giving the line

```
cout = 'c' // inputn
```

to a fortran compiler, and getting three error messages.

The compiler is Microsoft's F77 compiler for  
IBM PC

The error messages have been numeric expected and  
invalid expression (twice)

Both cout and inputn are defined as character\*15,  
although I tried making cout character\*20.

It seems to read inputn well enough, as I have been  
able to get around the problem using the format statements

I have no clue, Does anyone else

HELP!!!!

gino

---

Subject: Re: stupid problem in fortran  
Posted by [tj](#) on Tue, 23 Apr 1985 15:13:10 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Article-I.D.: utcs.614

Posted: Tue Apr 23 10:13:10 1985

Date-Received: Tue, 23-Apr-85 11:35:05 EST

References:

Reply-To: tj@utcs.UUCP (tj)

Organization: University of Toronto - General Purpose UNIX

Lines: 6

Xref: utcs net.micro.pc:3663 net.lang.f77:266

Summary:

well, I thought I knew... but I didn't... I recently had an experience with IBM mainframe fortran where I coded `cout = 'x' // cout` and instead of getting an x concatenated in front I got the first character replaced with the x. IBM documents this as a restriction, I call it and yours a bug!!!  
tjones

---

Subject: Re: stupid problem in fortran

Posted by [Anonymous](#) on Thu, 25 Apr 1985 20:28:57 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Originally posted by: kurtk&#64;tektronix.UUCP (Kurt Krueger)

Article-I.D.: tektroni.5318

Posted: Thu Apr 25 15:28:57 1985

Date-Received: Sat, 27-Apr-85 04:40:23 EST

References:

Reply-To: kurtk@tektronix.UUCP (Kurt Krueger)

Organization: Tektronix, Beaverton OR

Lines: 23

Xref: watmath net.micro.pc:3774 net.lang.f77:269

Summary:

This is in response to the fact that statements like:

```
var = 'x' // var
```

generally don't work as expected.

The following is an excerpt from a CDC fortran manual (and CDC has done a very good job of implementing the standard Fortran 77):

Character assignment statement

v = exp

.... None of the character positions being defined in v can be referenced in exp.

Personal comment: I too feel that the problem should be considered a bug, but it appears that the intent of the standard is to disallow this type of usage. I can also see an implementation headache for the compiler writer. The only solution is to use a temporary variable. A longer term solution would be to let your feelings be known to the ANSI committee so that f8x standard would allow this.

---

Subject: Re: stupid problem in fortran  
Posted by [ark](#) on Fri, 26 Apr 1985 16:13:44 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Article-I.D.: alice.3658  
Posted: Fri Apr 26 11:13:44 1985  
Date-Received: Sat, 27-Apr-85 05:19:30 EST  
References:  
Organization: Bell Labs, Murray Hill  
Lines: 21  
Xref: watmath net.micro.pc:3776 net.lang.f77:270

From the Fortran 77 standard, pages 10-2 to 10-3:

The form of a character assignment statement is:

v = e

where: v is the name of a character variable, character array element,  
or character substring  
e is a character expression

Execution of a character assignment statement causes the evaluation of the expression e and the assignment and definition of v with the value of e. None of the character positions being defined in v may be referenced in e.

Thus, a statement of the form

var = 'c' // var

violates the standard and an implementor is not obliged to allow it or even implement it sensibly.

---

Subject: Re: stupid problem in fortran  
Posted by [Anonymous](#) on Fri, 26 Apr 1985 22:54:30 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Originally posted by: pmontgom@sdcrdcf.UUCP (Peter Montgomery)

Article-I.D.: sdcrdcf.1930  
Posted: Fri Apr 26 18:54:30 1985  
Date-Received: Tue, 30-Apr-85 08:04:49 EDT  
References:  
Reply-To: pmontgom@sdcrdcf.UUCP (Peter Montgomery)  
Organization: System Development Corp. R+D, Santa Monica  
Lines: 25  
Xref: watmath net.micro.pc:3812 net.lang.f77:271  
Summary:

In article [tj@utcs.UUCP](#) (tj) writes:

>  
> well, I thought I knew... but I didn't... I recently had an experience with IBM  
> mainframe fortran where I coded `cout = 'x' // cout` and instead of getting  
> an x concatenated in front I got the first character replaced with the x.  
> IBM documents this as a restriction, I call it and yours a bug!!!  
> tjones

Section 10.4 of the FORTRAN standard, ANSI X3.9-1978, p. 10-2 defines the form of a character assignment statement to be

`v = e`

where v is a variable, array element, or substring, and e is a character expression. It also states that "None of the character positions being defined in v may be referenced in e". So this is indeed a restriction. If you don't obey the rules, you can't expect your program to work.

--

Peter Montgomery

{aero,allegra,bmcg,burdvax,hplabs,  
ihnp4,psivax,randvax,sdcsvox,trwrb}@sdcrdcf!pmontgom

Don't blame me for the crowded freeways - I don't drive.

---

Subject: Re: stupid problem in fortran  
Posted by [Anonymous](#) on Mon, 29 Apr 1985 15:06:58 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Originally posted by: b2&#64dalcs.UUCP (Bert Buckley)

Article-I.D.: dalcs.1497  
Posted: Mon Apr 29 11:06:58 1985  
Date-Received: Tue, 30-Apr-85 00:42:42 EDT  
References:  
Organization: Dalhousie University, Halifax, N.S., Canada  
Lines: 37

> This is in response to the fact that statements like:  
>  
> var = 'x' // var  
>  
> generally don't work as expected.  
>

I have two comments. First, when the discussion relates to f77 on Unix, I am never surprised when there is a failure to conform to the standard, and therefore would not be surprised by anything that f77 did to this statement.

But, of more relevance is what is supposed to happen within the standard. As has been pointed out in another article, overlapping string assignments are not permitted within the 1977 standard.

This question is however being considered by the Fortran Committee of ANSI, namely X3J3, but in a much broader context. Fortran 8x contains proposals for whole array operations and array sectioning. There is also a statement that says roughly that in the event of overlapping assignments (either for arrays or for strings or substrings), the effect is that of evaluation of the right hand side followed by assignment to the left. This is wonderful in principle, but it does not take a lot of effort to figure out that within subroutines this could have rather large implications, i.e. either that the compiler write opts for execution efficiency and ignores the possibility of overlap, or that he conform to the standard, implements all operations using temporary arrays or strings, and generates programs that grind along forever.

The Canadian Fortran Standards Committee has been following the

F8x proposals quite closely and has written to X3J3 and the International Fortran Standards Committee (ISO/TC97/SC22/WG5) expressing a number of concerns about proposals for F8x. Anyone wishing a copy of this report is welcome to one by sending me mail with a UUCP or ARPANET address; I will respond with a copy in TeX source form (which is reasonably readable without processing it with TeX).

---

---

Subject: Re: stupid problem in fortran  
Posted by [gino](#) on Wed, 01 May 1985 15:50:11 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Article-I.D.: sdchema.379  
Posted: Wed May 1 11:50:11 1985  
Date-Received: Sat, 4-May-85 00:43:14 EDT  
References:  
Reply-To: [gino@sdchema.UUCP](mailto:gino@sdchema.UUCP) (Eugene G. Youngerman)  
Organization: U.C. San Diego Chemistry Dept  
Lines: 20  
Xref: watmath net.micro.pc:3844 net.lang.f77:277  
Summary:

>> This is in response to the fact that statements like:  
>>  
>> var = 'x' // var  
>>  
>> generally don't work as expected.  
>>

I was the original poster  
The system I was running this on was Microsoft Fortran 77 for IBM compatible PC.

Also the original problem was

```
var = 'x' // var1
```

different variable names.

It seems that the PC version just doesn't support this stuff.  
These commands worked fine on the UNIX version I was using.

GINO

---

---

Subject: Re: stupid problem in fortran  
Posted by [neal](#) on Thu, 02 May 1985 16:48:28 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Article-I.D.: denelvx.17  
Posted: Thu May 2 12:48:28 1985  
Date-Received: Sat, 4-May-85 00:51:38 EDT  
References:  
Organization: Denelcor, Aurora, Colorados  
Lines: 15  
Xref: watmath net.micro.pc:3846 net.lang.f77:278

> This is wonderful in prin-  
> ciple, but it does not take a lot of effort to figure out that  
> within subroutines this could have rather large implications, i.e.  
> either that the compiler write opts for execution efficiency and  
> ignores the possibility of overlap, or that he conform to the  
> standard, implements all operations using temporary arrays or  
> strings, and generates programs that grind along forever.

Or the compiler is "smart" enough to figure out when it needs  
to do which.

Regards,  
Neal Weidenhofer  
"Nothin' ain't worth nothin' Denelcor, Inc.  
but it's free" !denelcor!neal

---

Subject: Re: stupid problem in fortran  
Posted by [tj](#) on Thu, 02 May 1985 19:49:06 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Article-I.D.: utcs.633  
Posted: Thu May 2 15:49:06 1985  
Date-Received: Thu, 2-May-85 16:27:03 EDT  
References:  
Reply-To: [tj@utcs.UUCP](mailto:tj@utcs.UUCP) (tj)  
Organization: University of Toronto - General Purpose UNIX  
Lines: 14  
Xref: utcs net.micro.pc:3761 net.lang.f77:273  
Summary:

Sure that is the rule, but if it is a rule, then let the compiler actually come out and tell me I am breaking the rules. The latest release (1.4) does tell me I am breaking the rules, version 1.3 does not. DEC fortran 77 on tops 10 does not tell me anything, but it does it right!

I can usually live with restrictions but this struck me as not too logical a restriction. PL/1 does it fine, some fortrons do it fine, why not the others.. You can sit by and live by the restrictions if you want, But I am going to bitch and complain till a REASONABLE solution is found. This may not be worth too much complaining but other little things like this in that past have been changed because someone asked for it.

t.jones

---

Subject: Re: stupid problem in fortran  
Posted by [ark](#) on Fri, 03 May 1985 15:08:37 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Article-I.D.: alice.3688  
Posted: Fri May 3 11:08:37 1985  
Date-Received: Sat, 4-May-85 01:50:09 EDT  
References:  
Organization: Bell Labs, Murray Hill  
Lines: 13  
Xref: watmath net.micro.pc:3850 net.lang.f77:279

- > Sure that is the rule, but if it is a rule, then let the compiler actually
- > come out and tell me I am breaking the rules. The latest release (1.4) does
- > tell me I am breaking the rules, version 1.3 does not. DEC fortran 77 on
- > tops 10 does not tell me anything, but it does it right!

That's not the spirit of Fortran.

Less flippantly, it is impossible in general to determine whether a given character assignment might have overlapping problems. For a compiler to do it "right" would mean that it would have to use temporary memory (and slow things down thereby) whenever there was the possibility of overlap. This would penalize some programs that conform to the standard to help other programs that do not.



Subject: Re: stupid problem in fortran  
Posted by [tj](#) on Sun, 05 May 1985 03:05:07 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Article-I.D.: utcs.638  
Posted: Sat May 4 23:05:07 1985  
Date-Received: Sun, 5-May-85 00:43:34 EDT  
References:  
Reply-To: [tj@utcs.UUCP](mailto:tj@utcs.UUCP) (tj)  
Organization: University of Toronto - General Purpose UNIX  
Lines: 5  
Xref: utcs net.micro.pc:3788 net.lang.f77:279  
Summary:

Just because it works on one machine doesn't mean its right!!!  
var = '' // var works on a DEC-10 but isn't right according to the  
standard as has been pointed out.....  
t.jones