

NOTICE

This document may contain information covered by one or more licenses, copyrights, and non disclosure agreements. Circulation of this document is restricted to holders of a license for the UNIX®, PWB/UNIX®, or Mini-UNIX® software system from Western Electric. Such license holders may reproduce this document for uses in conformity with the Unix license. All other circulation or reproduction is prohibited.

Address editorial material, payments, and software submission to
Melvin Ferentz
Box 8
The Rockefeller University
1230 York Avenue
New York, N.Y. 10021

Subscription requests and address changes should be addressed to
Armand Gazes
Box 8
The Rockefeller University
1230 York Avenue
New York, N.Y. 10021

*Trademark of Bell Laboratories

The City University of New York
Computer Center 565 West 57 Street, New York, N.Y. 10019 Telephone 212 977-8000

May 16, 1978

Professor Melvin Ferentz
Rockefeller University
Box 8
1230 York Avenue
New York, New York 10021

Dear Professor Ferentz:

There is a bug in Version 7 of the UNIX C Compiler. The source of the "cc" command is being packaged by Bell Labs with a call to the "alloc" routine of the C Compiler "-lc" library. The new "cc" command, which must be compiled with the "-ls" library, indirectly uses the "malloc" routine of that library. The routines "malloc" and "alloc", however, are essentially the same and the linker therefore generates "Multiply defined" errors.

To fix this problem, change the reference in "cc.c" from "alloc" to "malloc" such that "malloc" is used uniformly.

Our system has been running the "cc" command compiled this way for nearly a month without incident.

Yours truly,

Steven Eisen
Steven R. Eisen
PDP II System Manager

SE/cec

April 6, 1978
 Computer Graphics Laboratory
 New York Institute of Technology
 P.O. Box 170, Old Westbury, NY
 11568

UNIX news
 c/o Prof. Mel Ferentz
 Physics Dept.
 Brooklyn College of CUNY
 Brooklyn, NY 11210

Dear Mel:

By now it is well known that nargs(III) does not work in programmes with separated instruction and data spaces, and that furthermore, it cannot be made to work by using the mmpi instruction. Those of us that have tried to get it to go by writing a system entry which will read the instruction space of an I/D separated programme have discovered that the routine runs much too slowly to be of use. The version of nargs which follows works with separated I & D space (given a system entry readi(addr) which returns the contents of the given address in instruction space), and uses dynamic programming to run at a reasonable speed (usually). On the first call, the routine determines whether the programme is running with I/D separation or not. If not, it uses the algorithm of the currently distributed nargs. Otherwise, it first checks a table to see if it already knows how many arguments were passed to the calling routine, and if so returns. If not, it uses the 'classical' nargs algorithm, in conjunction with the readi system entry, and saves the result in the table before returning. Which table entry is used is determined by a first-in, first-out algorithm, modified so that a table entry which has been accessed since the last time we overwrote it (or tried to overwrite it) is not overwritten. (This can be viewed as a least-recently-used replacement strategy, with the time since last use stored to 1 bit precision). The size of the table (NLSIZE) is adjustable to taste. True speed addicts might want to rewrite this routine in assembly language.

```
#define NLSIZE 64
#define SIGSYS 12
#define JSRSD 04737
#define TSTI 05726
#define CMPI 022626
#define ADDI 062706
#define JMPI 0167
#define BRI 0777
struct nlist{
  int *pc;
  char nargs;
```

```
char touched;
};
struct(int integ;);
nargs(a){
  register *pc, inst, count;
  int savesys;
  struct nlist *np;
  static struct nlist nlist[NLSIZE], *nsweep;
  static notfirst, idsep;
  if(!notfirst){
    nsweep=nlist;
    /*
     * If we have I/D separation, the next
     * two calls to readi should return the
     * same value, at least one of which
     * will be different from notfirst, since
     * it gets changed between the calls.
     * SIGSYS must be ignored for the duration
     * because of the possibility that notfirst
     * is not a legal instruction space address
     */
    savesys=signal(SIGSYS, 1);
    if(readi(&notfirst)!=notfirst)
      idsep++;
    notfirst++;
    if(!idsep && readi(&notfirst)!=notfirst)
      idsep++;
    signal(SIGSYS, savesys);
  }
  pc=(a)[-2]; /* caller's r5 */
  pc=pc[1]; /* caller of caller's return addr */
  count=0;
  if(idsep){
    for(np=nlist;npl=nlist+NLSIZE;np++){
      if(np->pc==pc){
        np->touched=1;
        return(np->nargs&0377);
      }
    }
    while(nsweep->touched){
      nsweep->touched=0;
      if(++nsweep==nlist+NLSIZE)
        nsweep=nlist;
    }
    nsweep->pc=pc;
    if(readi(pc-2)==JSRSD)
      count=1;
    for(;;){
      inst=rcadi(pc);
      if(inst==JMPI){
```

PURDUE UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING
WEST LAFAYETTE, INDIANA 47907

March 31, 1978

```
pc.integ++readi(pc+1);
pc+=2;
}
else if(inst>>8==(BRI>>8)) {
pc.integ++readi(pc)<<8>>7;
pc++;
}
else
break;
}
if(inst==TSTI)
count+=1;
else if(inst==CMPI)
count+=2;
else if(inst==ADDI)
count++readi(pc+1)>>1;
nswEEP->nargs=count;
}
else {
if(pc[-2]==JSRSD)
count=1;
for(;;) {
inst = *pc;
if(inst==JMPI) {
pc.integ++pc|1;
pc+=2;
}
else if(inst>>8==(BRI>>8)) {
pc.integ++pc<<8>>7;
pc++;
}
else
break;
}
if(inst==TSTI)
count+=1;
else if(inst==CMPI)
count+=2;
else if(inst==ADDI)
count++pc[1]>>1;
}
return(count);
}
```

I hope this solution to the nargs problem can be of some use to UNIX sites with 11/70s and 11/45s.

Yours very truly,
Tom Bull
Tom Bull

Professor Melvin Ferentz
c/o CUNY/UCC
555 W. 57th Street
New York, New York 10019

Dear Professor Ferentz:

I would like to ask the UNIX user group of software developments in the area of TROFF to Versitec interfacing. At the present time we are running NEQN and NROFF outputs onto a Diablo 1610 typewriter terminal. Using various filters between NROFF and the Diablo, a relatively flexible high quality text output system has been achieved. In the future, we would like to extend our capabilities into the areas provided by TROFF but do not require the output quality of the GSI typesetter. We would like to further consider the concept of a TROFF to Versitec interface. Assistance from UNIX User Group members would be greatly appreciated.

Sincerely,

William R. Simmons

William R. Simmons
Manager, Digital Services

WRS/gb

VANDERBILT UNIVERSITY



NASHVILLE, TENNESSEE 37235

TELEPHONE (615) 322-7311

Electrical and Biomedical Engineering • Direct phone 322-2771

March 16, 1978

Dr. Melvin Ferentz
Box 8
The Rockefeller University
1230 York Avenue
New York, N. Y. 10021

Dear Dr. Ferentz:

We are at the present looking for a statistical package to run under Unix. So far we don't know if there is any compiler that will successfully compile BMDP or SPSS or any other available package, or if there is any package written for Unix. Any information will be greatly appreciated.

Sincerely,

Baruch Hamel

Baruch Hamel, Ph.D.
Research Associate

BH/ps

P.S. Would you be interested in forming a group of INGRES users within the Unix users group? Maybe you would like to discuss this possibility as well as its financial implications with Bob Epstein.