

---

Subject: Re: package build strange error  
Posted by [tojocky](#) on Wed, 03 Feb 2010 12:09:48 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

New interesting article about my problem:

Some innocent-looking programs are known to cause GCC to gobble preposterous amounts of memory, which could cause it to crash or abort after printing "Virtual memory exhausted". One particular case of such programs is when you initialize very large arrays. For example, to compile a source which initializes a char array of 300,000 elements requires more than 60MB(!) of memory. You should avoid such constructs in your programs.

Some programs require very large amounts of stack to compile. DJGPP programs have a fixed-size stack that is by default 256KB (512KB in DJGPP v2.02 and later). If the compiler, cc1.exe or cc1plus.exe, doesn't have enough stack to compile a program, it will overflow its stack and crash, or hang, or die with "Internal compiler error". You can enlarge the stack size of any DJGPP program by running the stubedit program, like this:

```
stubedit cc1.exe minstack=1024k
```

I recommend to enlarge the maximum stack size of cc1.exe to at least 1024K bytes and that of cc1plus.exe to at least 1.5MB. Some people report that they needed to enlarge both the heap of CWSDPMI and the stack of the C++ compiler to make such problems go away.

For a program that you wrote, another work-around for the cases where a program crashes due to failure of CWSDPMI to allocate more RAM is to use an alternative algorithm for sbrk, by putting the following somewhere in your program:

```
#include <crt0.h>
```

```
int _crt0_startup_flags = _CRT0_FLAG_UNIX_SBRK;
```

Note that the Unix algorithm for sbrk might cause trouble in programs that install hardware interrupts handlers.

source: [http://www.delorie.com/djgpp/v2faq/faq6\\_6.html](http://www.delorie.com/djgpp/v2faq/faq6_6.html)

Question: is possible to implement this?

regards, Ion Lupascu (tojocky)

---