

docgen is a document extractor for C++ code written by
Mark Stupar for aips++ in April 1992.

It is a simple filter which reads from standard in and writes document
lines to standard out. The intention is to have another program read
the extracted doc file and write a man page or texinfo file.

The program assumes the following:

any line of code beginning with "/*." is considered a document line
and will be written to standard out. The maximum line size is assumed
to be 125 bytes.

sort order	code	name
0	/*.*n	<name of class>
1	/*.*l	<language>
2	/*.*s	<synopsis>
3	/*.*t	<time/date of last mod>
4	/*.*o	<one-line description>
5	/*.*r	<responsible/"owner">
7	/*.*b0	<base class> (private)
	/*.*b1	(protected)
	/*.*b2	(public)
10	/*.*fc	<friend class>
24	/*.*z	<begins with /*. but doesn't fit scheme; output>
6	/*.*p	<paragraph>
11	/*.*d0	<data> (private)
	/*.*d1	<data> (protected)
	/*.*d2	<data> (public)
14	/*.*c0	<constructor> (private)
	/*.*c1	<constructor> (protected)
	/*.*c2	<constructor> (public)
17	/*.*k0	<destructor> (private)
	/*.*k1	<destructor> (protected)
	/*.*k2	<destructor> (public)
20	/*.*m0	<member function> (private)
	/*.*m1	<member function> (protected)
	/*.*m2	<member function> (public)
23	/*.*ff	<friend function>

/*.* => one line item
/*.* => multi-line item
 terminated by a
/*.*-

see A2_Timer.h for an example.

```
// -----  
//  
// A2_Timer.h - include file for A2_Timer class  
//  
// History:  
//  
// 21apr92 mjs Adapted from TI's COOL/lite (03/22/91 version).  
// -----  
// Copyright (C) 1991 Texas Instruments Incorporated.  
//  
// Permission is granted to any individual or institution to use, copy, modify,  
// and distribute this software, provided that this complete copyright and  
// permission notice is maintained, intact, in all copies and supporting  
// documentation.  
//  
// Texas Instruments Incorporated provides this software "as is" without  
// express or implied warranty.  
// -----  
  
#ifndef A2_TIMER  
#define A2_TIMER  
#include <sys/types.h>
```

```

#include <sys/timex.h>
#include <sys/time.h>
#include <sys/resource.h>

//.*n A2_Timer
//.*l Sun CC (2.1)
//.*l GNU g++ (v2)
//.*l CenterLine CC (Nov 1991)
//.*s #include "A2_Timer.h"
//.*t 21apr92
//.*o Provide timing code for performance evaluation.
//.*r AIPS++ Basic Libraries
//.*p Description
//. The A2_Timer class provides an interface to system timing, allowing
//. a C++ program to record the time between a reference point (mark) and
//. now. The class uses the system time(2) interface to provide time
//. resolution at either millisecond or microsecond granularity, depending
//. on operating system support and features. Since the time duration is
//. stored in a 32-bit word, the maximum time period before rollover
//. occurs is about 71 minutes.
//.-

class A2_Timer {
private:
    rusage  usage0;
    timeb   real0;

    //.*d0 rusage usage0;
    //.* rusage structure at last mark.
    //.-
    //.*d0 timeb real0;
    //.* elapsed real time at last mark.
    //.-

public:
    A2_Timer () {mark();}
    //.*c2 A2_Timer()
    //.* Creates an instance of the A2_Timer class with the mark set to
    //.* creation time.
    //.-

    void mark ();
    //.*m2 void mark()
    //.* Sets the reference time to now.
    //.-

    long user ();
    //.*m2 long user()
    //.* Returns the number of milliseconds spent in the user process since
    //.* the last reference point (mark).
    //.-

    long system ();
    //.*m2 long system()
    //.* Returns the number of milliseconds spent in the operating system
    //.* since the last reference point (mark).
    //.-

    long all ();
    //.*m2 long all()
    //.* Returns the number of milliseconds spent in the user process and the
    //.* operating system since the last reference point (mark).
    //.-

    long real ();
    //.*m2 long real()
    //.* Returns the number of milliseconds of wall clock time since the last
    //.* reference point (mark).
    //.-

    long user_usec ();
    //.*m2 long user_usec()
    //.* Returns the number of microseconds spent in the user process since
    //.* the last reference point (mark).

```

```
///  
-
```

```
    long system_usec ();  
///  
+m2 long system_usec()  
///  
Returns the number of microseconds spent in the operating system  
since the last reference point (mark).  
///  
-
```

```
    long all_usec ();  
///  
+m2 long all_usec()  
///  
Returns the number of microseconds spent in the user process and the  
operating system since the last reference point (mark).  
///  
-  
};
```

```
#endif // define A2_TIMER
```