The UNIX[1] time will run out in 2038!
by Richard A. Painter, P.E.

© Copyright 1996 - 2005 Painter Engineering, Inc.

8470 Swan Rd.
Black Forest, CO 80908
719 495 7054
painter@ieee.org

Granting of limited rights

Permission is granted to use, duplicate, and publish this text and program as long as it includes the Painter Engineering, Inc. address block, descriptive text, code, and this limited rights reference.

This brief and program is in response to issues of the YEAR 2000 computer problems and one aspect of date and time for UNIX systems.

The UNIX time will run out in 2038! Once the magic time arrives the time may revert to 13 Dec 1901 or reset to the UNIX birth date of 1 Jan 1970 depending on the underlying hardware and implementation.

All UNIX systems (I know of no exceptions at this time) rely on a basic system call function called *time()* to obtain the time. This function returns a (signed) long integer (4 bytes) of seconds since the EPOCH. The EPOCH is defined as the birth of UNIX, that is 1 Jan 1970. The internal structure *time_t* (signed long integer) is used throughout many time-related functions and data structures, including the file systems.

Without vendor support it is unlikely a feasible solution could be developed to work around this limit. There are many, many UNIX systems deployed and still in use even though the vendors don't exist or don't support them anymore. There are so many references to the *time_t* structure in the UNIX operating system, commands and applications (UNIX utilities) source code that it would not be portable without changing most versions from most vendors. For those without UNIX source there is no chance to modify the required structures, functions and utilities.

The attached program demonstrates the limitations of the UNIX time functions and internal time representations.

The arithmetic for determining the last date and time for this *time_t* structure follows:

max seconds in a long	= 2147483647 seconds
	= 35791394 minutes, 7 seconds
	= 596523 hours, 14 minutes, 7 seconds
	= 24855 days, 3 hours, 14 minutes, 7 seconds
	= 68 years, 18 days, 3 hours, 14 minutes, 7 seconds

Therefore, 68 years plus 1970 yields 2038. The point in time that this will happen is Tue 19 Jan 2038 at 03:14 and 7 sec. AM GMT. Please note that this accounts for the leap year days too.

This C program demonstrates the boundaries and effects of the start and end dates used in UNIX with the *time_t* structure. The program output sample is included at the end of the program.

[1] UNIX is a trademark of AT&T.

```
/*
    +-------------------------------------------------------+
    | Richard A. Painter, P.E.            painter@ieee.org  |
    | Painter Engineering, Inc.                             |
    | 8470 Swan Rd.                                         |
    | Black Forest, CO 80908                                |
    |                                                       |
    | http://painterengineering.com       719 495 7054      |
    |                                                       |
    | Computer and Software Forensics                       |
    | Systems & Software Engineering + Consulting + Audits  |
    | LAN WAN Networking + Real-time Transaction Processing |
    | Year 2000 Solutions + UNIX + MS Windows + Security    |
    +-------------------------------------------------------+

      (c) Copyright 1996 - 2005 Painter Engineering, Inc.
*/
#include <stdio.h>
#include <limits.h>
#include <time.h>

main()
{
    time_t maxt;

    maxt = 0L;
    printf("Start of UNIX GMT time (%ld, 0x%08X) is %s",
            maxt, maxt, asctime(gmtime(&maxt)));

    maxt = LONG_MAX;    /* 2147483647L */
    printf("End of UNIX GMT time is (%ld, 0x%08X) is %s",
            maxt, maxt, asctime(gmtime(&maxt)));

    maxt++;
    printf("End of UNIX GMT time plus 1 sec. (%ld, 0x%08X) is %s",
            maxt, maxt, asctime(gmtime(&maxt)));

    maxt = 0L;
    maxt--;
    printf("Start of UNIX GMT time minus 1 sec. (%ld, 0x%08X) is %s",
            maxt, maxt, asctime(gmtime(&maxt)));
}
/*
cc -O -o maxtime maxtime.c
cc -O -o maxtime maxtime.c
*/

/*          the output on a Data General AViiON is
Start of UNIX GMT time (0, 0x00000000) is Thu Jan  1 00:00:00 1970
End of UNIX GMT time is (2147483647, 0x7FFFFFFF) is Tue Jan 19 03:14:07
2038
End of UNIX GMT time plus 1 sec. (-2147483648, 0x80000000) is Fri Dec
13 20:45:52 1901
```

```
Start of UNIX GMT time minus 1 sec. (-1, 0xFFFFFFFF) is Wed Dec 31
23:59:59 1969
*/
```