



“CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2016”

Multidisciplinario

21 y 22 de abril de 2016, Cortazar, Guanajuato, México

ISBN: 978-607-95635

Solución MIPS para el problema del 2038 para las basadas en POSIX

Jesús Antonio Alvarez-Cedillo¹, Juan Carlos Herrera Lozada¹ y Jacobo Sandoval-Gutierrez²

¹Instituto Politécnico Nacional, CIDETEC, México

²UAM-Lerma, México

Emails: jaalvarez@ipn.mx, jozada@ipn.mx, jacobosandovag@gmail.com

Resumen: El problema del Año 2038, es también conocido también como el error Y2K38, representa una limitación técnica para los sistemas POSIX, afecta el dato utilizado para representar el tiempo y podría ocasionar el mal funcionamiento de computadoras, Smartphone y equipos electrónicos. El problema se presentaría a las 03:14:07 (UTC) del 19 de enero de 2038, y tiene su origen en una variable interna utilizada para contar el tiempo en los sistemas de 32 bits es un problema en hardware y software, ya que algunos programas no podrían asignar suficientes dígitos para representar el año, después de la fecha antes mencionada, el reloj se desbordará y devolverá datos erróneos. Algunos sistemas que integran esta tecnología son los satélites, sondas espaciales, sistemas de navegación, plantas eléctricas, bombas atómicas, controladores de misiles, controles y en muchos otros campos. Así que después de la fecha antes mencionada, estos sistemas tendrán fallos. En este artículo se propone una solución basada en un el conteo del día Juliano a partir del cual se genera una fecha base (Epoch time) y se mantiene la cuenta de la fecha actual desde el momento en que se produce el cambio al bit de signo, esta solución se implemento en lenguaje ensamblador para MIPS.

Palabras clave--- 2k38; Arquitectura de computadoras; EPOCH;MIPS

1. INTRODUCCIÓN

Cuando la fecha del sistema en las computadoras alcance el 19 de Enero de 2038 a las 3:14:07 horas, un error se presentará. Cualquier computadora, programa, servidor o aplicación que se ejecute con la base de un sistema de 32 bits, fallará inevitablemente, al menos que sea su código arreglado y actualizado. Este problema es conocido como el problema del año 2038, y es una teoría que fue probada recientemente cuando el video de Gangnam style de Psy en Youtube, excedió las dos millones de reproducciones. El problema del año 2038, específicamente afecta a aquellos sistemas que trabajan en una

1 | “Congreso Internacional de Investigación e Innovación 2016” Multidisciplinario, 21 y 22 de abril de 2016. México



“CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2016”

Multidisciplinario

21 y 22 de abril de 2016, Cortazar, Guanajuato, México

ISBN: 978-607-95635

arquitectura de 32 bits. Los sistemas de 32 bits usan 4 bytes, lo que significa que pueden usar 4 Gb de memoria. En comparación un sistema de 16 bits utiliza 2 bytes y uno de 64 bits utiliza 8 bytes. Estos sistemas almacenan la memoria y liberan procesos usando dígitos binarios, representados como 0 o 1, y el número total de dígitos que un sistema de 32 bits es capaz de representar es 4, 294, 967, 295 valores. Debido a que la mitad de los valores son negativos, y la otra mitad positivos, el rango total va de -2, 147, 483, 648 hasta 2, 147, 483, 647. Esto significa que solamente existe un número limitado de almacenamiento para guardar los unos y ceros. El formato estándar que se utiliza de 4 bytes para el sistema de hora Posix, asume el inicio del tiempo como el 1 de Enero de 1970, a las 12:00 horas. Cuando el reloj alcance la cuenta a las 3:14:07 el martes 19 de Enero de 2038, un total de 2, 147, 483,647 habrán pasado desde el primero de Enero de 1970.

Una posible consecuencia se da al dejar de funcionar los relojes de las computadoras operan con arquitecturas de 32bits, también colapsarían las bases de datos, protocolos de comunicación, Smartphones, redes de transporte, equipos industriales, electrodomésticos entre otras de uso cotidiano que manejamos hoy en día.

1.1 DIA JULIANO

El calendario juliano es el antecesor del calendario gregoriano, que actualmente usamos, y su funcionamiento esta basado en el movimiento del sol en el universo para medir el tiempo, fue implementado en el año 46 a. C., se adoptó al principio en los países europeos , cuando se promulga la reforma gregoriana de 1582 por el papa Gregorio XIII. Algunos países cuya religión es ortodoxa, se mantuvo hasta principios del siglo XX. El año 153 a. C. se tomo como inicio del año el 1 de enero, consta de 365 días divididos en 12 meses, excepto los años bisiestos que tienen 366 días, y añaden un día adicional al mes de febrero.

El calendario juliano tiene presente los años bisiestos, los seculares, tiene una precisión en la cual solo se comete un error de 3 días cada 400 años. Además renombró el mes de Quintil, como Julius (Julio), en su honor. La manera de contar los días siguió la tradición romana hasta que los visigodos introdujeron la costumbre de numerar los días, que no sería oficial hasta que la adoptó Carlomagno. Tal como se muestra en la Tabla 1.



“CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2016”

Multidisciplinario

21 y 22 de abril de 2016, Cortazar, Guanajuato, México

ISBN: 978-607-95635

Tabla 1. Distribución de los meses y los días.

Mundo Clásico		En la actualidad	
Mes	Dias	Mes	Dias
Mártium	31	Mártium	31
April	30	April	30
Máium	31	Máium	31
Júnium	30	Júnium	30
Quintil	30	Júlium	31
Sextil	30	Augústum	31
Septémber	31	Septémber	30
Octóber	30	Octóber	31
Novémber	31	Novémber	30
Decémber	30	Decémber	31
Januárium	31	Januárium	31
Februárium	30 y 31 bisiestos	Februárium	28 y 29 bisiestos

La fecha juliana, representa es el número de días y fracción transcurridos desde el mediodía del 1 de Enero del año 4713 a. C. La fecha juliana es una cuenta continua de días y fracciones contados desde un punto inicial fijo. En este momento, la parte entera corresponde al día ; la fracción indica la fracción de día transcurrida desde el último mediodía (0,5 es media noche). En computación, una variable de punto flotante de doble precisión (64 bits) con la cual se permite representar una fecha juliana con 1 milisegundo de precisión. Para el caso de MIPS es posible extrapolar ambas tablas para construir un solo registro de segundos de 32 bits, tal como se presenta en el problema 2k38.

A pesar de contar con 24 años de ventaja, se ha comenzado a pensar cómo evitar este problema. Es importante mencionar que todas las computadoras que trabajan con sistemas operativos en 64bits no se verán afectadas con este problema. La solución propuesta se basa en actualizar la los algoritmos de los equipos para evitar un vacío tecnológico que podría detener en parte el funcionamiento de internet, entre otras cosas [1]. A diferencia de la propuesta que plantea la necesidad de invertir un trillón de dólares para renovar los equipos y reducir las consecuencias [2].En el siguiente apartado se describe la solución generada para el problema Y2038 utilizando lenguaje ensamblador sobre un procesador MIPS programado en el simulador Mars 4.5.

Como se muestra en la Figura 1, las computadoras no serán capaces de distinguir entre la hora y fecha real, y el año de 1901 debido al desbordamiento de un entero, significa que el contador se quedó sin bits para utilizar y comienza a enviar números negativos. Para algunas computadoras, este problema solo afectaría en mostrar una fecha incorrecta a la



“CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2016”

Multidisciplinario

21 y 22 de abril de 2016, Cortazar, Guanajuato, México

ISBN: 978-607-95635

actual, pero también podría desconfigurar a las computadoras y programas que utilizan el reloj interno del sistema para obtener mediciones precisas, o incluso provocaría que todo el software falle[3]. Versiones modernas con el sistema operativo de 32 bits han resuelto este problema cambiando el registro del tiempo a un entero con signo de 64 bits para contar los segundos, pero algunos dispositivos antiguos o aquellos que funcionan con versiones antiguas de software, corren el riesgo de fallar impredeciblemente. De esta manera, los sistemas afectados necesitarán ser actualizados para sí evitar futuros problemas[4].

```

Binary : 01111111 11111111 11111111 11111111
Decimal : 2147483647
Date : 2038-01-19 03:14:07 (UTC)
Date : 2038-01-19 03:14:07 (UTC)

Binary : 10000000 00000000 00000000 00000000
Decimal : -2147483648
Date : 1901-12-13 20:45:52 (UTC)
Date : 2038-01-19 03:14:08 (UTC)

```

Figura 1. Problema del 2038.

2. Metodología

Los equipos que basan su control del tiempo en el sistema Posix de Unix, el cual usa el tiempo Epoch como control del tiempo son los mostrados en la Tabla 2. El tiempo Epoch es la hora actual medida en número de segundos desde el Epoch Unix.

Epoch 0 es enero 1 1970 00:00:00 GMT (ISO 8601: 1970-01-01T00:00:00Z). No se usan segundos bisiestos en el tiempo Unix.

Tabla 2. Sistemas Operativos Afectados.



“CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2016”

Multidisciplinario

21 y 22 de abril de 2016, Cortazar, Guanajuato, México

ISBN: 978-607-95635

Table with 3 columns: POSIX, Compatibles pero no POSIX, Utilizan POSIX. Rows include A/UX, AIX, BSD, DSPnano, HP-UX, INTEGRITY, IRIX, LINXOS, MACOSX /DARWIN, MINIX, MPE/IX, QNX, SOLARIS, BEOS, FREBSD, GNU/LINUX, CONTIKI, NETBSD, NUCLEUS RTOS, OPENBSD, OPENSOLARIS, PIKEOS, RTOS, RTEMS, SANOS, SKYOS, ANDROID, IOS, WINDOWS PHONE, UBUNTU TOUCH, FIREFOX OS, BADA, BLACKBERRY, SYMBIAN, WINDOWS.

La propuesta parte del manejo del registro de tiempo de 32 bits donde el problema se encuentra en el ultimo bit del registro, bit que maneja el signo del valor entero contenido. Al llevar un conteo de los segundos transcurridos sobre la fecha inicial hasta el máximo valor del entero con signo, el conteo pondrá en 1 el bit más significativo, cambiando a un valor negativo el contenido en el registro, arrojando una fecha errónea. La solución planteada en este documento para este problema es trabajar con un nuevo registro también de 32 bits, el cual tendrá la tarea de seguir con el conteo de la nueva fecha, una vez que se alcance el límite del registro, es decir, que partir de un incremento sobre el nuevo registro, el registro original de la cuenta de la fecha comenzara nuevamente su conteo.

Una vez alcanzado el valor máximo positivo del registro se establece una nueva cuenta la cual comenzara ahora sobre el año 2038 y no sobre 1970, el conteo del registro de la fecha iniciara sobre el día 19 del mes de enero con la hora siguiente correspondiente al último valor alcanzado en el registro, esto para mantener un nuevo conteo a partir de la fecha del 1 de enero del 2038.

La programación parte con la configuración de los registro para la simulación del año 2038, es decir, el registro de la fecha en simulación comienza 14 segundos antes del límite del valor li \$s1,0x7FFFFFF0.

Como siguiente parte y clave importante del manejo de las fechas, es verificar el tiempo en el que se encuentra actualmente los registros, si estamos sobre las fechas iniciales antes del 2038 o después.



“CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2016”

Multidisciplinario

21 y 22 de abril de 2016, Cortazar, Guanajuato, México

ISBN: 978-607-95635

La Figura 2 muestra el diagrama general de la verificación de esta validación. Como se mencionó anteriormente, cuando el bit 32 cambie de 0 a 1, habrá un desbordamiento y comenzarán a obtenerse números negativos que provocarán que el sistema piense que el año actual es 1901 cuando en realidad debe ser 2038.

Por esta razón se propone como solución temporal a este problema y evitar que sea necesario migrar todos los sistemas que basan el control del tiempo de un registro de 32 bits a 64 bits por dos registros de 32 bits, uno como contador y el otro como registro de estado para actualizar la fecha de referencia que usa el tiempo Epoch, con esto, contar con otros 68 años, donde es posible que ya los equipos basados en este modelo de referencia del tiempo sean sustituidos por una nueva tecnología.

Binary 1	: 00000000 00000000 00000000 00000000
Binary 0	: 01111111 11111111 11111111 11111111
Decimal	: 2147483647
Date	: 2038-01-19 03:14:07 (UTC)
Date	: 2038-01-19 03:14:07 (UTC)
Binary 1	: 00000000 00000000 00000000 00000001
Binary 0	: 00000000 00000000 00000000 00000000
Decimal	: 2147483648
Date	: 2038-01-19 03:14:08 (UTC)
Date	: 2038-01-19 03:14:08 (UTC)

Figura 2. Propuesta del problema 2038.

La manera de hacer el cambio de referencia es utilizando el estado del bit más significativo como bandera para activar el cambio de referencia y que el contador de tiempo continúe funcionando de la misma manera pero iniciando la cuenta el 19 de enero de 2038 a las 3:14:07 am. Utilizando la tabla anterior, calculamos una fecha cercana a la del 19 de enero de 2038 para simular el momento en que se presente el desbordamiento del registro de 32 bits, como se muestra a continuación.

Usando de referencia el valor de 2, 147, 483, 640, se divide entre el número de segundos que tiene un año obteniendo, con lo cual se obtiene la siguiente expresión:

$$\text{Años}_{32 \text{ bits}} = \frac{\text{segundos en el registro de 32bits}}{\text{segundos en un año Juliano}} \quad (1)$$



“CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2016”

Multidisciplinario

21 y 22 de abril de 2016, Cortazar, Guanajuato, México

ISBN: 978-607-95635

En base a (1), para la fecha de creación de este articulo se obtuvo:

Años_32 bits} = 2147483640 / 31556926 = 68.051103583409

De manera similar, para determinar el año en el que se producirá el error, desde el 1 de enero de 1970, se le suma la cantidad de años obtenidos , usando la expresión (1) de la siguiente forma:

Año_error = Inicio1 Epoch + Años_32bits (2)

Con lo que se obtiene:Año_error = 1970 + 68 = 2038

Para calcular el mes adicional, a partir del residuo de (1) se multiplica por el numero de meses en el año:

Mes_faltantes = Residuo Años_32 bits * numero de meses (3)

Con lo que se obtiene:

Mes_faltantes = 0.51103583409 * 12 = 0.613243000908

Como el mes no llega a un número entero, se considera que se trata de enero, ya que se realiza un ajuste debido a los años bisiestos que hay entre 1970 y 2038.

Se toman los meses obtenidos en (3) y se multiplican por el número aproximado de días que tiene un mes y con esto se obtienen cuantos días han pasado después del mes:

Días_faltantes = Residuo del mes * promedio de día mensual (4)

Con lo que se obtiene:

Día_faltantes = 0.613243000908 * 30.44 = 18.66711694763952

Nuevamente se hace un ajuste de un día al valor obtenido en (4), ya que este valor afecta de manera importante los años bisiestos en el transcurso del tiempo, por lo que el día es 19.

Debido al ajuste que se hace al día, se toma del nuevo residuo de (4) , de los decimales para completar el día, y obtener las horas transcurridas.

Horas_faltantes = (1 - residuo Día_faltantes) * Horas Dia (5)



“CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2016”

Multidisciplinario

21 y 22 de abril de 2016, Cortazar, Guanajuato, México

ISBN: 978-607-95635

Obtenemos:

$$\text{Horas}_{faltantes} = (1 - 0.66711694763952) * 24 = 3.23333333333$$

Una vez obtenidas las horas se hace el cálculo de los minutos multiplicando los decimales (5) por el numero de minutos en una hora.

$$\text{Minutos}_{faltantes} = \text{Residuo } \text{Horas}_{faltantes} * \text{minutos en una hora} \quad (6)$$

Se obtuvo:

$$\text{Minutos}_{faltantes} = 0.23333333333 * 60 = 14.0$$

Como no hay decimales se determina que hay exactamente 0 segundos en el momento de la cuenta Epoch tomada, por lo que la fecha resultante queda: 19 – 01 – 2038 3:14:00. Como ya se determinó a que fecha corresponde el número Epoch, se procede a programar en ensamblador, utilizando el programa MARS versión 4.5 como compilador, para simular el comportamiento del reloj próximo a la fecha del desbordamiento.

El diagrama de la Figura 3, muestra el proceso donde inicialmente se verifica el registro que hace referencia al año 2038 si este es mayor a cero se activa la bandera que indica que los años siguientes comenzaran en el 2038 de no ser así, se verifica que la sustracción del registro de la fecha con el registro del límite sea menor a cero, de no ser así se activa la bandera del año 2038 y el reset del registro de la fecha. Una vez realizado este proceso, se continua con el cálculo de la fecha, ya sea con inicio de 1970 o 2038. El cálculo de la fecha se realiza de la siguiente manera. El algoritmo 1 muestra el procedimiento utilizado.



“CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2016”
Multidisciplinario

21 y 22 de abril de 2016, Cortazar, Guanajuato, México

ISBN: 978-607-95635

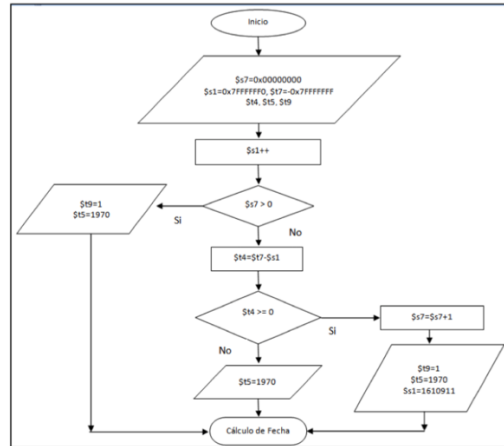


Figura 3. Diagrama de verificación de fecha.

Algoritmo 1 Código del Programa desarrollado en la aplicación Mars Versión 4.5

Inicio
Se establece la Referencia Inicial Epoch 1-1-1970
BanderaAc=0
Se establece la hora inicial
Se ajusta reinicio de segundos
Para todo segundo #hora1:
 Leer hora de sistema
 Si la hora registrada es igual a 2147483647
 Asigna nueva hora al al registro Epoch
 Indica que existió un acarreo
 Modifica El formato de fecha
 AsignaBanderaAc=1
 En caso contrario
 Continúa Procedimiento
 Si Bandera Ac=0
 Muestra Fecha Posix
 En caso contrario
 Muestra fecha Posix modificada
Fin



“CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2016”

Multidisciplinario

21 y 22 de abril de 2016, Cortazar, Guanajuato, México

ISBN: 978-607-95635

3 IMPLEMENTACION

Como ya se determinó a que fecha corresponde el número Epoch, se procede a programar en ensamblador, utilizando el programa MARS versión 4.5 como compilador, para simular el comportamiento del reloj próximo a la fecha del desbordamiento.

De acuerdo a la Algoritmo 2, se establece la referencia inicial de la cuenta del tiempo utilizando un registro independiente para segundos, minutos, horas, días, meses y años. También se guarda la hora inicial obtenida así como el numero en formato Epoch en registros independientes para comenzar la simulación del comportamiento del reloj del sistema segundos antes del desbordamiento que ocurrirá.

Algoritmo 2. Referencia inicial y hora inicial para la simulación.

```

.data . align 2
.globl main .text
main:
#-----Referencia Inicial Epoch empezando la cuenta a partir del 1-01-1970 0:0:0
li $s6, 0
li $s7, 0
li $t8, 0
li $t9, 1
li $k0, 1
li $k1, 1970
#-----Hora Inicial----- li $t0, 2147483640
li $s0, 0
li $s1, 14
li $s2, 3
li $s3, 19
li $s4, 1
li $s5, 2038
li $t1, 0
#Referencia para reinicio de segundos      Muestra fecha Posix modificada

```



“CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2016”

Multidisciplinario

21 y 22 de abril de 2016, Cortazar, Guanajuato, México

ISBN: 978-607-95635

Posteriormente de inicializar los registros, se realiza la actualización de los registros para que la estructura muestre la presentación de la fecha y la hora. Como se aprecia en el Algoritmo 3, primero se despliegan los años luego los meses y los días, separados por un guion “- “. Después se despliegan las horas, minutos y segundo separados entre sí por dos puntos “:.”. La parte más importante del programa sucede cuando se habilita la instrucción de la Algoritmo 4.

Esta instrucción se habilita cuando el contador llega al número máximo de cuenta y que después se desbordará, para evitar eso se lanza una subrutina que interrumpe el proceso normal de ejecución del programa y entra al segmento de código etiquetado “hora 2”.

Algoritmo 3 Estructura de la visualización de la Fecha y hora en el programa..

```

hora1:- 

```



“CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2016”

Multidisciplinario

21 y 22 de abril de 2016, Cortazar, Guanajuato, México

ISBN: 978-607-95635

```

li $a0, ':'
syscall
li $v0, 1
move $a0, $s1 syscall
li $v0, 11
li $a0, ':'
syscall
li $v0, 1
move $a0, $s0 syscall

li $v0, 11
li $a0, ''
syscall
li $v0, 35 #mostramos entero add $a0, $t0, $zero
syscall

```

Algoritmo 4 Instrucción que habilita La condición de cambio en la estructura del contador Epoch.

```

Beq $t0, 2147483647, hora2

#Se habilita la bandera que indica el cambio de referencia

```

En el Algoritmo 5 se muestra la parte del código que se encarga de controlar los cambios de la hora y la fecha dependiendo de los segundo y del tiempo Epoch que ya se registró. Siempre será el mismo proceso por lo que es un ciclo que se repite cada vez.

Algoritmo 5 Parte del programa que controla el comportamiento del cambio del tiempo.

```

hora11:
add $s0, $s0, 1
sle $t1, $s0, 59 #Segundos que cambia a 0
seq $t2, $t1, 0 #minutos que incrementan si segundos es > 59
add $s1, $s1, $t2
sle $t1, $s1, 59 #minutos que cambia a 0

```



“CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2016”

Multidisciplinario

21 y 22 de abril de 2016, Cortazar, Guanajuato, México

ISBN: 978-607-95635

mul \$s1,\$s1,\$t1

seq \$t2,\$t1,0 #horas que incrementan si minutos es > 59 add \$s2,\$s2,\$t2

sle \$t1,\$s2,23 #horas que cambia a 0

mul \$s2,\$s2,\$t1

seq \$t2,\$t1,0 #días que incrementan si horas es > 24 add \$s3,\$s3,\$t2

li \$v0,11 #Entre

li \$a0,'n'

syscall

add \$t0,\$t0,1 b hora1

Cuando la cuenta Epoch llega al número máximo de cambios positivos disponibles, el programa entra a la subrutina “hora2”, en la cual se cambian los valores iniciales cargados en los registros de la referencia de hora inicial por la del 19 de Enero de 2038 a las 3:14:0. Además la cuenta Epoch se reinicia al valor 8, con el propósito de conservar la misma relación de los cambios con las fechas considerando cada cambio de bit como un incremento de un segundo, y así funcionará la función del tiempo de la misma forma que lo hace con la referencia del 1 de Enero de 1970. Observe el algoritmo 6.

Algoritmo 6 Segmento de código que realiza la modificación a la referencia inicial del contador Epoch.

hora2:

#——Cambio Referencia Inicial Epoch empezando la cuenta a partir del 19-01-2038 3:14:0———

li \$s6, 0

li \$s7, 14

li \$t8, 3

li \$t9, 19

li \$k0, 1

li \$k1, 2038

#

li \$t0,8 #Se reinicia el contador Epoch b hora11

Al correr el programa se empieza a obtener la hora próxima al desbordamiento, y se observa que se comporta de manera correcta ya que al llegar a la hora máxima, se reinicia el contador Epoch a 8 pero no se modifica la fecha actual, por lo que se resuelve el problema definido en este documento, observe la Figura 4.



“CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2016”
Multidisciplinario

21 y 22 de abril de 2016, Cortazar, Guanajuato, México

ISBN: 978-607-95635

4 CONCLUSIONES

La herramienta MARS 4.5 permite crear la simulación de un procesador MIPS de 32 bits, para la solución propuesta al problema o bug del 2038, no se utilizaron las funciones correspondientes al tiempo ya que estas trabajan con dos registros de 32 bits parte alta y baja como contador unitario del tiempo y el manejo del tiempo es en milisegundos, cuando el problema viene afectado sobre procesadores POSIX donde trabajan con un solo registro de 32 bits con signo, donde cada bit es representado como un segundo. Es importante mencionar que durante la investigación no se propuso la solución de utilizar un registro sin signo, ya que el signo negativo es indispensable para cálculo de las fechas inferiores a 1970 hasta 1900.

Mars Messages	Run I/O
2038-1-19 3:14:0	011111111111111111111111111111111111000
2038-1-19 3:14:1	011111111111111111111111111111111111001
2038-1-19 3:14:2	011111111111111111111111111111111111010
2038-1-19 3:14:3	011111111111111111111111111111111111011
2038-1-19 3:14:4	011111111111111111111111111111111111100
2038-1-19 3:14:5	011111111111111111111111111111111111101
2038-1-19 3:14:6	011111111111111111111111111111111111110
2038-1-19 3:14:7	011111111111111111111111111111111111111
2038-1-19 3:14:8	000000000000000000000000000000000001001
2038-1-19 3:14:9	000000000000000000000000000000000001010
2038-1-19 3:14:10	000000000000000000000000000000000001011
2038-1-19 3:14:11	000000000000000000000000000000000001100
2038-1-19 3:14:12	000000000000000000000000000000000001101
2038-1-19 3:14:13	000000000000000000000000000000000001110
2038-1-19 3:14:14	000000000000000000000000000000000001111

Figura 4. Ejecución del programa que muestra el resultado obtenido.

Durante la obtención de la fecha del registro, el cálculo puede variar dependiendo de las conversiones realizadas, ya que el año puede variar en la cantidad de días, horas, minutos o segundos contenidos en este, para evitar esto es necesario basarse en el día juliano para obtener la precisión esperada.

Finalmente se espera que durante los próximos 23 años, se agreguen mejoras a esta alternativa, siendo pertinente dar un seguimiento cercano a los trabajos que se publiquen en el futuro.



“CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2016”

Multidisciplinario

21 y 22 de abril de 2016, Cortazar, Guanajuato, México

ISBN: 978-607-95635

REFERENCIAS

[1] Glow-worm PJ, Year 2038 bug problem , The pharmaceutical Journal, 2013, A royal Pharmaceutical Society Publication, pp 201-210.

[2] Vishal Singh, Perna Chaudhary, International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-2, Issue-1, October 2012.

[3] Carrington, Damian. Was Y2K bug a boost, BBC News. Archived from the original on 22 April 2004. Retrieved 19 September 2009.

[4] IEE, The Open Group Base Specifications Issue 6 IEEE Std 1003.1, 2004 Edition. IEEE and The Open Group. The Open Group.2004. Retrieved 7 March 2008.

[5] Diomidis Spinellis, Code quality: the open source perspective, Effective software development series in Safari Books Online 2009.