

## Ryzom - Bug # 899

<b>Status:</b>	New	<b>Priority:</b>	Normal
<b>Author:</b>	rti	<b>Category:</b>	Build
<b>Created:</b>	05/14/2010	<b>Assignee:</b>	
<b>Updated:</b>	01/08/2011	<b>Due date:</b>	

**Subject:** The undefined reference to `\_\_sync\_bool\_compare\_and\_swap\_4' problem

### Description

Hi.

I just looked a bit further into the "undefined reference to `\_\_sync\_bool\_compare\_and\_swap\_4'" problem people are getting lately due to #874.

```
int main(int p_argc, char const** p_argv) {
    char val = 0;
    __sync_bool_compare_and_swap(&val, 0, 1);
    return 0;
}
```

I succeeded building this code with the following commands:

```
gcc/g++ -m32 -march=i486 ...
```

```
gcc/g++ -m64 ...
```

The building fails with the famous undefined reference to `\_\_sync\_bool\_compare\_and\_swap\_4' with the following commands:

```
gcc/g++ -m32 -march=i386 ...
```

I think the reason that brocifer was not able to compile the code (<http://dev.ryzom.com/boards/17/topics/1648>) is, that he only set the CFLAGS and not the CXXFLAGS variable. Since the call to `__sync_bool_compare_and_swap` resides in a C++ header, setting CFLAGS had simply no effect.

So I think the solution to this problem should be:

- add the `-march` value on `[[BuildForLinux]]`
- ensure that cmake generated build scripts contain the right `-march` setting on gcc

### History

#1 - 05/15/2010 09:47 am - vl

Perhaps a more portable solution should be to detect (how?) if the function is available or not, and use the `asm` if the function is not available.

#2 - 05/15/2010 04:22 pm - rti

**Only versions of gcc >= 4.1 provide this function**

<http://stackoverflow.com/questions/2118992/how-do-i-use-gcc-builtin-sync-bool-compare-and-swap-in-g-on-macosx>

Could be checked using the following defines provided by GCC:

```
#define __GNUC__ 4
#define __GNUC_MINOR__ 2
```

**If the instruction is not available on the architecture**

...but the compiler version is >= 4.1, the call is redirected:

"Not all operations are supported by all target processors. If a particular operation cannot be implemented on the target processor, a warning will be generated and a call an external function will be generated. The external function will carry the same name as the builtin, with an additional suffix `\_n'` where n is the size of the data type." (taken from <http://gcc.gnu.org/onlinedocs/gcc-4.1.1/gcc/Atomic-Builtins.html>)

That's where the `__sync_bool_compare_and_swap_4` comes from. So implementing this function using the ASM Macro might be a way... But then again it does not make much sense, because GCC "thinks" the current architecture has no XCHG instruction, as a result the code would force the ASM to contain XCHG.

#3 - 05/15/2010 06:04 pm - vl

Agree that i386 is quite old and every processor should have xchg so i think we can definitively change the to use at least i486

#4 - 05/18/2010 10:25 am - vl

- Target version set to Version 0.8.0

#5 - 06/25/2010 03:40 pm - rti

Just saw that the new PCH script does stuff like that:

```
IF(gcc_compiler_version MATCHES "4\\.[0-9]\\.[0-9]")
```

Just wanted to mention... On that basis we could decide whether to use the ASM or the gcc built-in, or, just complain that minimum version of GCC is 4.2 :)

But still the target processor arch needs to be checked / forced.

#6 - 07/02/2010 12:58 am - rti

Just saw another thing... in CMakeModules/nel.cmake, there is some stuff which sets CMAKE\_SYSTEM\_PROCESSOR using uname (~line 140).

Maybe this could be a way to force a minimum architecture for x86?

#7 - 10/15/2010 07:38 pm - kerval

Perhaps this link could help :)

<http://code.google.com/p/iphonedevonlinux/issues/detail?id=10>

#8 - 01/08/2011 06:44 pm - kervalá

- *Target version deleted (Version 0.8.0)*