Memory alignment

Too many problems popped up because of unnoticed misaligned memory access in kernel code lately. Therefore the alignment fixup is now unconditionally configured in for SA11x0 based targets. According to Alan Cox, this is a bad idea to configure it out, but Russell King has some good reasons for doing so on some f***ed up ARM architectures like the EBSA110. However this is not the case on many design I'm aware of, like all SA11x0 based ones.

Of course this is a bad idea to rely on the alignment trap to perform unaligned memory access in general. If those access are predictable, you are better to use the macros provided by include/asm/unaligned.h. The alignment trap can fixup misaligned access for the exception cases, but at a high performance cost. It better be rare.

Now for user space applications, it is possible to configure the alignment trap to SIGBUS any code performing unaligned access (good for debugging bad code), or even fixup the access by software like for kernel code. The later mode isn't recommended for performance reasons (just think about the floating point emulation that works about the same way). Fix your code instead!

Please note that randomly changing the behaviour without good thought is real bad - it changes the behaviour of all unaligned instructions in user space, and might cause programs to fail unexpectedly.

To change the alignment trap behavior, simply echo a number into /proc/cpu/alignment. The number is made up from various bits:

bit	behavior when set
0	A user process performing an unaligned memory access will cause the kernel to print a message indicating process name, pid, pc, instruction, address, and the fault code.
1	The kernel will attempt to fix up the user process performing the unaligned access. This is of course slow (think about the floating point emulator) and not recommended for production use.
2	The kernel will send a SIGBUS signal to the user process performing the unaligned access.

Note that not all combinations are supported - only values 0 through 5. (6 and 7 don't make sense).

For example, the following will turn on the warnings, but without fixing up or sending SIGBUS signals:

```
echo 1 > /proc/cpu/alignment
```

You can also read the content of the same file to get statistical information on unaligned access occurrences plus the current mode of operation for user space code.

Nicolas Pitre, Mar 13, 2001. Modified Russell King, Nov 30, 2001.