cmse_nonsecure_entry

The tracking issue for this feature is: #75835

The TrustZone-M feature is available for targets with the Armv8-M architecture profile (`thumbv8m` in their target name). LLVM, the Rust compiler and the linker are providing support for the TrustZone-M feature.

One of the things provided, with this unstable feature, is the `cmse_nonsecure_entry` attribute. This attribute marks a Secure function as an entry function (see section 5.4 for details). With this attribute, the compiler will do the following: * add a special symbol on the function which is the `__acle_se_` prefix and the standard function name * constrain the number of parameters to avoid using the Non-Secure stack * before returning from the function, clear registers that might contain Secure information * use the `BXNS` instruction to return

Because the stack can not be used to pass parameters, there will be compilation errors if: * the total size of all parameters is too big (for example more than four 32 bits integers) * the entry function is not using a C ABI

The special symbol `__acle_se_` will be used by the linker to generate a secure gateway veneer.

```
#![feature(cmse_nonsecure_entry)]

#[no_mangle]
#[cmse_nonsecure_entry]
pub extern "C" fn entry_function(input: u32) -> u32 {
    input + 6
}
```

```
$ rustc --emit obj --crate-type lib --target thumbv8m.main-none-eabi function.rs
$ arm-none-eabi-objdump -D function.o

00000000 <entry_function>:
   0:   b580            push    {r7, lr}
   2:   466f            mov     r7, sp
   4:   b082            sub     sp, #8
   6:   9001            str     r0, [sp, #4]
   8:   1d81            adds    r1, r0, #6
   a:   460a            mov     r2, r1
   c:   4281            cmp     r1, r0
   e:   9200            str     r2, [sp, #0]
  10:   d30b            bcc.n   2a <entry_function+0x2a>
  12:   e7ff            b.n     14 <entry_function+0x14>
  14:   9800            ldr     r0, [sp, #0]
  16:   b002            add     sp, #8
```

```
18:   e8bd 4080       ldmia.w sp!, {r7, lr}
1c:   4671            mov     r1, lr
1e:   4672            mov     r2, lr
20:   4673            mov     r3, lr
22:   46f4            mov     ip, lr
24:   f38e 8800       msr     CPSR_f, lr
28:   4774            bxns    lr
2a:   f240 0000       movw    r0, #0
2e:   f2c0 0000       movt    r0, #0
32:   f240 0200       movw    r2, #0
36:   f2c0 0200       movt    r2, #0
3a:   211c            movs    r1, #28
3c:   f7ff fffe       bl      0 <_ZN4core9panicking5panic17h5c028258ca2fb3f5E>
40:   defe            udf     #254    ; 0xfe
```