```
# [...]

# EXTENSION 2019-06-12

    #-----------------------------------------------------------------------
    # The interrupt 3 IDT gate DPL must be redifed to User Level in order for
    # the User module processes to call the int3 instruction.
    #-----------------------------------------------------------------------
    carica_gate  3  breakpoint  LIV_UTENTE

# EXTENSION 2019-06-12

# [...]

# EXTENSION 2019-06-12

    # initialize IDT gate for the TIPO_B (breakpoint() primitive) interrupt
    carica_gate  TIPO_B  a_breakpoint  LIV_UTENTE

# EXTENSION 2019-06-12

# [...]

# SOLUTION 2019-06-12

#---------------------------------------------------------------------------
# When the breakpoint() primitive is called, if everything goes well the calling
# process is placed in the system global breakpoint descriptor queue. Otherwise,
# in case of error, the calling process is aborted. In either case the state
# of the current process has to be saved and the state of a new process must be
# loaded.
#---------------------------------------------------------------------------
a_breakpoint:
    .cfi_startproc
    .cfi_def_cfa_offset 40
    .cfi_offset rip, -40
    .cfi_offset rsp, -16
    call salva_stato             # save current process state
    call c_breakpoint            # call C++ implementation
    call carica_stato            # load new process state
    iretq                        # return from interrupt
    .cfi_endproc

# SOLUTION 2019-06-12

# [...]

#---------------------------------------------------------------------------
# Interrupt 3 - Breakpoint Exception handler
# The breakpoint exception handler must be redefined in order for the user
# process which called the breakpoint() primitive to be rescheduled. When the
# breakpoint exception is handled, if the breakpoint was inserted by the
# breakpoint() primitive both processes must be place in the system ready
# processes queue and a new process must be scheduled. If it is a breakpoint
# exception not related to the breakpoint() primitive, the calling process
# must be aborted a new one must be scheduled. In either case the current
# process state must be saved and a new process state must be loaded.
#---------------------------------------------------------------------------
breakpoint:
    .cfi_startproc
    .cfi_def_cfa_offset 40
    .cfi_offset rip, -40
    .cfi_offset rsp, -16
    call salva_stato

# SOLUTION 2019-06-12

    movq $3, %rdi                # interrupt type
    movq $0, %rsi                # error type
    movq %rsp, %rdx              # address contained in %rsp
```

```
    call c_breakpoint_exception      # call C++ handler

# SOLUTION 2019-06-12

    call carica_stato                # load new process state
    iretq
    .cfi_endproc

# [...]
```