Problem

- `schedule()` and `switch_to()` macro
  - `schedule()` is a tricky function. It returns with a different stack
  - If old and new `schedule()` are compiled differently by GCC (different registers are allocated, stack is different due to local variables etc.) → corruption and crash upon return
  - Possible solution – ensure the return to a proper version of `schedule()`
    - Make the value of RIP register part of the saved and restored context

switch_to() macro – pre 4.9

```
asm volatile(SAVE_CONTEXT
  "movq %%rsp,%P[threadrsp](%[prev])\n\t" /* save RSP */
  "movq %P[threadrsp](%[next]),%%rsp\n\t" /*restore RSP */
  "call __switch_to\n\t”
  “.globl thread_return\n”
  “thread_return:\n”
  …
```

Changed switch_to()

```
asm volatile(SAVE_CONTEXT
  "movq %%rsp,%P[threadrsp](%[prev])\n\t" /* save RSP */
  "movq %P[threadrsp](%[next]),%%rsp\n\t" /* restore RSP */
  "movq $thread_return,%P[threadrip](%[prev])\n\t"/*save RIP*/
  "pushq %P[threadrip](%[next])\n\t" /* restore RIP */
  "jmp __switch_to\n\t"
  ".globl thread_return\n"
  "thread_return:\n\t"
  …
```