

Relatório P1 SO II

Gustavo Kundlatsch, Paola de Oliveira, Pedro Souza

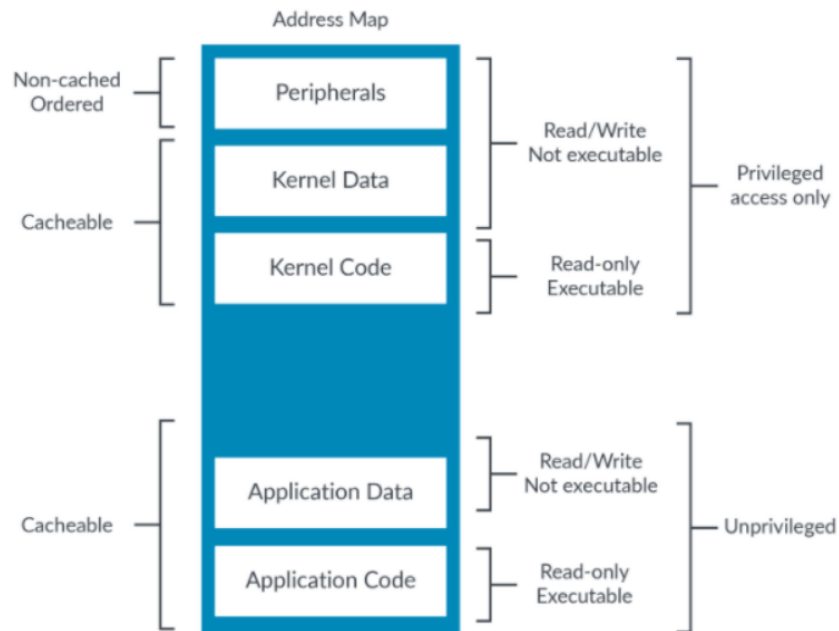
28 de setembro de 2022

O objetivo do grupo era aplicar o diff do exercício anterior, duplicar os arquivos da arquitetura ArmV8 como ponto de partida e ir mudando passo a passo cada uma das partes da implementação.

O modelo de memória escolhido foi o da imagem abaixo, disponível na página [1]:

Armv8 Memory Model

A typical Arm memory model looks like this



Todavia, encontramos problemas ao tentar compilar o EPOS com o cross compiler `aarch64-linux-gnu-gcc`. As flags utilizadas no makedefs foram as seguintes:

```

1 ifeq ($(MMOD),raspberry_pi3)
2 cortex_CC_FLAGS      := -mcpu=cortex-a53 -mabi=lp64 -Wno-attributes
3 cortex_EMULATOR     = qemu-system-aarch64 -M raspi3 -cpu
   ↪ cortex-a53 -smp 4 -m 1G -serial null -serial mon:stdio
   ↪ -nographic -no-reboot -kernel
4 cortex_AS_FLAGS      := -mcpu=cortex-a53
5 cortex_LD_FLAGS      :=
6 cortex_DEBUGGER       := gdb
7 cortex_FLASHER       = $(TLS)/eposflash/eposflash-$(MMOD) -d
   ↪ /dev/ttyACM0 -f
8 cortex_MAGIC         := --nmagic
9 cortex_CODE_NAME     := .init
10 cortex_DATA_NAME     := .data
11 cortex_IMG_SUFFIX    := .bin
12 endif

```

Mas o seguinte erro de compilação não foi resolvido:

```

1 ine5424/include/architecture/mmu.h:153:9: error: Phy_Addr does
   ↪ not name a type
2 153 |         Phy_Addr physical(Log_Addr addr) { return addr; }

```

Tentamos fazer uma versão mínima compilável removendo todos os assembly de Army7 do código e substituindo por nop (versão enviada), mas mesmo assim o erro persistiu.

Referências

- [1] Tom Zhao. Armv8 memory model. https://blog.tomzhao.me/wp-content/uploads/2021/08/Armv8_Memory_Model_aaeddabb09a9e4244a82dfc34230b5f2_180821_2356.pdf, 2021. [Online; accessed 04-Janeiro-2022].