

# Relatório P5 SO II

Gustavo Kundlatsch, Paola de Oliveira, Pedro Souza

28 de setembro de 2022

O objetivo do grupo era entregar o P5 funcional, um microkernel protegido, com syscalls, o application loader e múltiplas aplicações. Infelizmente, surgiram alguns problemas que serão discutidos em partes nas seções seguintes, separadas por critério de avaliação da grading table.

# 1 Command Line

Para executar o loader em linha de comando, implementamos um shell script que realiza as operações que precisariam ser escritas manualmente para carregar os programas:

```
make veryclean
rm img/app_loader*
rm img/loader.img
make APPLICATION=hello
make APPLICATION=app_loader
./bin/eposmkbi . ./img/loader.img ./img/app_loader ./img/hello
/usr/bin/arm-none-eabi-objcopy -O binary img/loader.img img/app_loader.bin
make APPLICATION=app_loader run
```

O script limpa o ambiente, compila os apps que serão carregados, utiliza o eposmkbi para gerar a imagem de boot do loader e copia ela para um binário para executá-lo. O script funciona mas os erros de compilação no loader (que serão discutidos na próxima seção) causam problemas.

O script está disponível no arquivo `loader.sh`, que precisa de permissão de execução para ser rodado. Outros aplicativos podem ser carregados com ele simplesmente adicionando mais linhas de compilação com o make e as imagens na linha do eposmkbi.

## 2 Loader

O loader foi implementado como um app (`app_loader`), que aloca segmentos e cria uma task para cada app que foi enviado ao loader através do script:

```
for (int app_size = *_argv; app_size; extras += app_size/4,
app_size = *reinterpret_cast<int*>(_argv)) {
    ELF * ini_elf = reinterpret_cast<ELF *>(++_argv);
    ...
    Elf32_Addr ini_entry = ini_elf->entry();
    int ini_segments = ini_elf->segments();
    Elf32_Addr ini_code = ini_elf->segment_address(0);
    int ini_code_size = ini_elf->segment_size(0);
    Elf32_Addr ini_data = 0xffffffff;
    int ini_data_size = 0;
    if(ini_elf->segments() > 1) {
        for(int i = 1; i < ini_elf->segments(); i++) {
            if(ini_elf->segment_type(i) != PT_LOAD)
                continue;
            if(ini_elf->segment_address(i) < ini_data) {
                ini_data = ini_elf->segment_address(i);
            }
            ini_data_size += ini_elf->segment_size(i);
        }
    }
    ini_data_size = _SYS::MMU::align_page(ini_data_size);
    EPOS::S::Segment * cs = new EPOS::S::Segment(ini_code_size);
    _SYS::CPU::Log_Addr cs_addr = c_task->address_space()->attach(cs);
    ini_elf->load_segment(0, cs_addr);
    c_task->address_space()->detach(cs, cs_addr);
    ...
    for(int j = 1; j < ini_elf->segments(); j++){
        if(ini_elf->segment_size(j) > 0){
            ini_elf->load_segment(j, aux_ds_addr);
            aux_ds_addr += ini_elf->segment_size(j);
        }
    }
    c_task->address_space()->detach(ds, ds_addr);
    typedef int (Main)();
    Main * n_main = reinterpret_cast<Main *>(ini_entry);
    _SYS::CPU::Log_Addr h_ini_code = reinterpret_cast<void *>(ini_code);
    _SYS::CPU::Log_Addr h_ini_data = reinterpret_cast<void *>(ini_data);
    new EPOS::S::Task(cs, ds, n_main, h_ini_code, h_ini_data);
    cout << "App created" << endl;
}
...
```

```

/usr/bin/aarch64-linux-gnu-ld: app_loader.cc: na função "main":
app_loader.cc:(.text.main+0x7f80): referência não definida para "EPOS::S::System:: heap"
/usr/bin/aarch64-linux-gnu-ld: app_loader.cc:(.text.main+0x7fb4): referência não definida para "EPOS::S::System:: heap"
/usr/bin/aarch64-linux-gnu-ld: app_loader.cc:(.text.main+0x1080): referência não definida para "EPOS::S::System:: heap"
/usr/bin/aarch64-linux-gnu-ld: app_loader.cc:(.text.main+0x1084): referência não definida para "EPOS::S::System:: heap"
/usr/bin/aarch64-linux-gnu-ld: app_loader.cc:(.text.main+0x1298): referência não definida para "EPOS::S::Segment::Segment(unsigned int, EPOS::S::MMU_Common<11u, 11u, 14u>::Flags)"
app_loader.cc:(.text.main+0x1298): realocação truncada para caber: R_AARCH64_CALL26 contra símbolo não definido "EPOS::S::Segment::Segment(unsigned int, EPOS::S::MMU_Common<11u, 11u, 14u>::Flags)"
/usr/bin/aarch64-linux-gnu-ld: app_loader.cc:(.text.main+0x12b0): referência não definida para "EPOS::S::Address_Space::attach(EPOS::S::Segment*)"
app_loader.cc:(.text.main+0x12b0): realocação truncada para caber: R_AARCH64_CALL26 contra símbolo não definido "EPOS::S::Address_Space::attach(EPOS::S::Segment*)"
/usr/bin/aarch64-linux-gnu-ld: app_loader.cc:(.text.main+0x1340): referência não definida para "go_user_mode"
/usr/bin/aarch64-linux-gnu-ld: app_loader.cc:(.text.main+0x1344): referência não definida para "go_user_mode"
/usr/bin/aarch64-linux-gnu-ld: app_loader.cc:(.text.main+0x13f0): referência não definida para "EPOS::S::Thread::constructor_prologue(unsigned int)"
app_loader.cc:(.text.main+0x13f0): realocação truncada para caber: R_AARCH64_CALL26 contra símbolo não definido "EPOS::S::Thread::constructor_prologue(unsigned int)"
/usr/bin/aarch64-linux-gnu-ld: app_loader.cc:(.text.main+0x1458): referência não definida para "EPOS::S::Thread::constructor_epilogue(EPOS::S::CPU_Common::Address_unsigned long*, unsigned int)"
app_loader.cc:(.text.main+0x1458): realocação truncada para caber: R_AARCH64_CALL26 contra símbolo não definido "EPOS::S::Thread::constructor_epilogue(EPOS::S::CPU_Common::Address_unsigned long*, unsigned int)"
/usr/bin/aarch64-linux-gnu-ld: app_loader.cc:(.text.main+0x14d0): referência não definida para "EPOS::S::Address_Space::Address_Space()"
app_loader.cc:(.text.main+0x14d0): realocação truncada para caber: R_AARCH64_CALL26 contra símbolo não definido "EPOS::S::Address_Space::Address_Space()"
/usr/bin/aarch64-linux-gnu-ld: app_loader.cc:(.text.main+0x1594): referência não definida para "EPOS::S::Address_Space::attach(EPOS::S::Segment*, EPOS::S::CPU_Common::Address_unsigned long*)"
app_loader.cc:(.text.main+0x1594): realocação truncada para caber: R_AARCH64_CALL26 contra símbolo não definido "EPOS::S::Address_Space::attach(EPOS::S::Segment*, EPOS::S::CPU_Common::Address_unsigned long*)"
/usr/bin/aarch64-linux-gnu-ld: app_loader.cc:(.text.main+0x151c): referência não definida para "EPOS::S::Address_Space::attach(EPOS::S::Segment*, EPOS::S::CPU_Common::Address_unsigned long*)"
app_loader.cc:(.text.main+0x151c): realocação truncada para caber: R_AARCH64_CALL26 contra símbolo não definido "EPOS::S::Address_Space::attach(EPOS::S::Segment*, EPOS::S::CPU_Common::Address_unsigned long*)"
/usr/bin/aarch64-linux-gnu-ld: app_loader.cc:(.text.main+0x166c): referência não definida para "EPOS::S::Task::task_count"
app_loader.cc:(.text.main+0x166c): realocação truncada para caber: R_AARCH64_CALL26 contra símbolo não definido "EPOS::S::Task::task_count"
/usr/bin/aarch64-linux-gnu-ld: app_loader.cc:(.text.main+0x1670): referência não definida para "EPOS::S::Task::task_count"
app_loader.cc:(.text.main+0x1670): realocação truncada para caber: R_AARCH64_CALL26 contra símbolo não definido "EPOS::S::Task::task_count"
/usr/bin/aarch64-linux-gnu-ld: app_loader.cc:(.text.main+0x169c): referência não definida para "EPOS::S::System:: heap"
app_loader.cc:(.text.main+0x169c): realocação truncada para caber: R_AARCH64_CALL26 contra símbolo não definido "EPOS::S::System:: heap"
/usr/bin/aarch64-linux-gnu-ld: app_loader.cc:(.text.main+0x16a0): referência não definida para "EPOS::S::System:: heap"
app_loader.cc:(.text.main+0x16a0): realocação truncada para caber: R_AARCH64_CALL26 contra símbolo não definido "EPOS::S::System:: heap"
/usr/bin/aarch64-linux-gnu-ld: app_loader.cc:(.text.main+0x1710): referência não definida para "EPOS::S::System:: heap"
app_loader.cc:(.text.main+0x1710): realocação truncada para caber: R_AARCH64_CALL26 contra símbolo não definido "EPOS::S::System:: heap"
/usr/bin/aarch64-linux-gnu-ld: app_loader.cc:(.text.main+0x1ae0): referência não definida para "EPOS::S::Segment::Segment(unsigned int, EPOS::S::MMU_Common<11u, 11u, 14u>::Flags)"
app_loader.cc:(.text.main+0x1ae0): realocação truncada para caber: R_AARCH64_CALL26 contra símbolo não definido "EPOS::S::Segment::Segment(unsigned int, EPOS::S::MMU_Common<11u, 11u, 14u>::Flags)"
/usr/bin/aarch64-linux-gnu-ld: app_loader.cc:(.text.main+0x1ef4): referência não definida para "EPOS::S::Segment::Segment(unsigned int, EPOS::S::MMU_Common<11u, 11u, 14u>::Flags)"
app_loader.cc:(.text.main+0x1ef4): realocação truncada para caber: R_AARCH64_CALL26 contra símbolo não definido "EPOS::S::Segment::Segment(unsigned int, EPOS::S::MMU_Common<11u, 11u, 14u>::Flags)"
/usr/bin/aarch64-linux-gnu-ld: app_loader.cc:(.text.main+0x2038): referência não definida para "EPOS::S::Stub_Task::current"
app_loader.cc:(.text.main+0x2038): realocação truncada para caber: R_AARCH64_CALL26 contra símbolo não definido "EPOS::S::Stub_Task::current"
/usr/bin/aarch64-linux-gnu-ld: app_loader.cc:(.text.main+0x2044): referência não definida para "EPOS::S::Stub_Task::current"
app_loader.cc:(.text.main+0x2044): realocação truncada para caber: R_AARCH64_CALL26 contra símbolo não definido "EPOS::S::Stub_Task::current"
make[3]: *** [makefile:8: app_loader] Erro 1

```

Figura 1: Erro do loader.

Infelizmente não pudemos verificar a corretude do código, pois não conseguimos corrigir alguns erros durante sua execução:

Acreditamos que esses erros de "referência não definida para "EPOS::S: ... " estejam sendo causados por algum problema de namespace. Da iteração do P4 para o P5, trocamos a maneira como estávamos compilando e criamos um arquivo `framework/main.h` para o projeto:

```

#ifdef __FRAMEWORK_MAIN_H
#define __FRAMEWORK_MAIN_H

#define EXPORT(X) typedef _SYS::X X;

__BEGIN_API

__USING_UTIL

EXPORT(CPU);
EXPORT(Handler);
EXPORT(System);
EXPORT(Application);

typedef _SYS::Stub_Thread Thread;
typedef _SYS::Stub_Task Task;
typedef _SYS::Stub_Address_Space Address_Space;
typedef _SYS::Stub_Segment Segment;
typedef _SYS::Stub_Mutex Mutex;
typedef _SYS::Stub_Semaphore Semaphore;
typedef _SYS::Stub_Condition Condition;

```

```
typedef _SYS::Stub_Clock Clock;  
typedef _SYS::Stub_Chronometer Chronometer;  
typedef _SYS::Stub_Alarm Alarm;  
typedef _SYS::Stub_Delay Delay;  
typedef _SYS::Stub_Shared_Segment Shared_Segment;  
  
__END_API  
  
#endif
```

O grupo acha que essa mudança pode ter afetado algum componente do loader de maneira a causar os erros, mas não foi capaz de identificar com precisão o problema ou resolvê-lo.

### 3 Shared System Objects

Para os shared system objects, adicionamos a classe Shared\_Segment:

```
class Shared_Segment: public Segment
{
private:
    int _port;
    typedef List<Shared_Segment> SS_List;

public:
    static SS_List _shared_segments;

public:
    Shared_Segment(int port, unsigned int bytes);

    static Shared_Segment * using_port(int port);

    void set_port(int port) {_port = port;}
    int get_port() {return _port;}
};
```

Um segmento compartilhado possui uma porta a qual pode se conectar, e os segmentos são gerenciados através de uma lista. O controle que verifica se o usuário irá utilizar um segmento que já existe através de uma porta ou criar um segmento novo é feito diretamente na classe do agent:

```
switch(method()) {
    case Message::SHARED_SEGMENT_CREATE: {
        int port;
        unsigned int bytes;
        get_params(port, bytes);
        Shared_Segment * shared_seg = Shared_Segment::using_port(port);
        if (!shared_seg) {
            shared_seg = new (SYSTEM) Shared_Segment(port, bytes);
        }
        result(reinterpret_cast<long>(shared_seg));
        db<Agent>(TRC) << "Stub Shared Segment CREATE" << endl;
    } break;
    case Message::SHARED_SEGMENT_PORT: {
        Shared_Segment * shared_seg = reinterpret_cast<Shared_Segment*>(id());
        int port = shared_seg->get_port();
        result(port);
        db<Agent>(TRC) << "Stub Shared Segment PORT" << endl;
    } break;
    default:
        db<Agent>(TRC) << "Shared segment failed" << endl;
}
```

```
        break;  
    }
```

Pelo que foi testado, a implementação parece estar funcionando corretamente.

## 4 Integration

A integração é um dos critérios que não foi bem sucedido por conta do problema do loader. Outros aplicativos quando testados individualmente pareceram estar funcionando, pelo menos parcialmente, mas quando o requisito se torna a integração com o loader e o funcionamento em conjunto de múltiplos aplicativos carregados de uma vez só, os erros de compilação não permitem que o objetivo seja alcançado. Os possíveis motivos para essa falha estão descritos na seção do loader.



## 5 Diff

Caso alguma dúvida a respeito do código tenha surgido, o diff foi disponibilizado no seguinte link: <https://gist.github.com/kundlatsch/dad0c9236a85d08c71a30eedc20829ef>