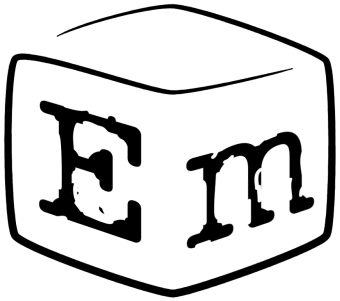


Реализация механизмов виртуальной памяти для x86 архитектуры

Курсовая работа

Антон Бондарев
Научный руководитель

Глеб Ефимов
345 группа

Май 2012

Цели

- Добавить поддержку виртуальной памяти в ОСРВ Embox
- Обеспечить линейное адресное пространство для процессов
- Обеспечение защиты доступа к памяти

Свойства виртуальной памяти

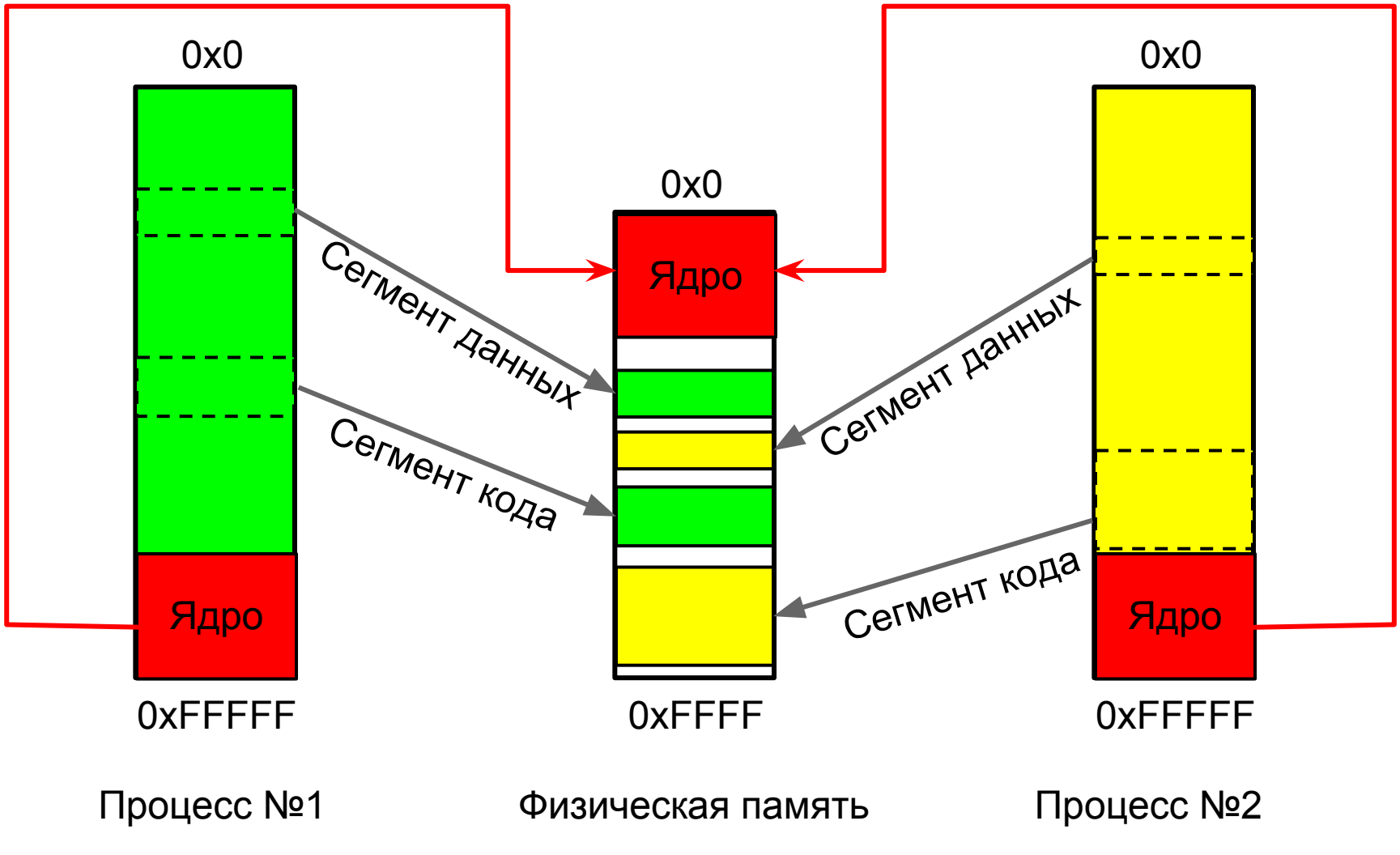
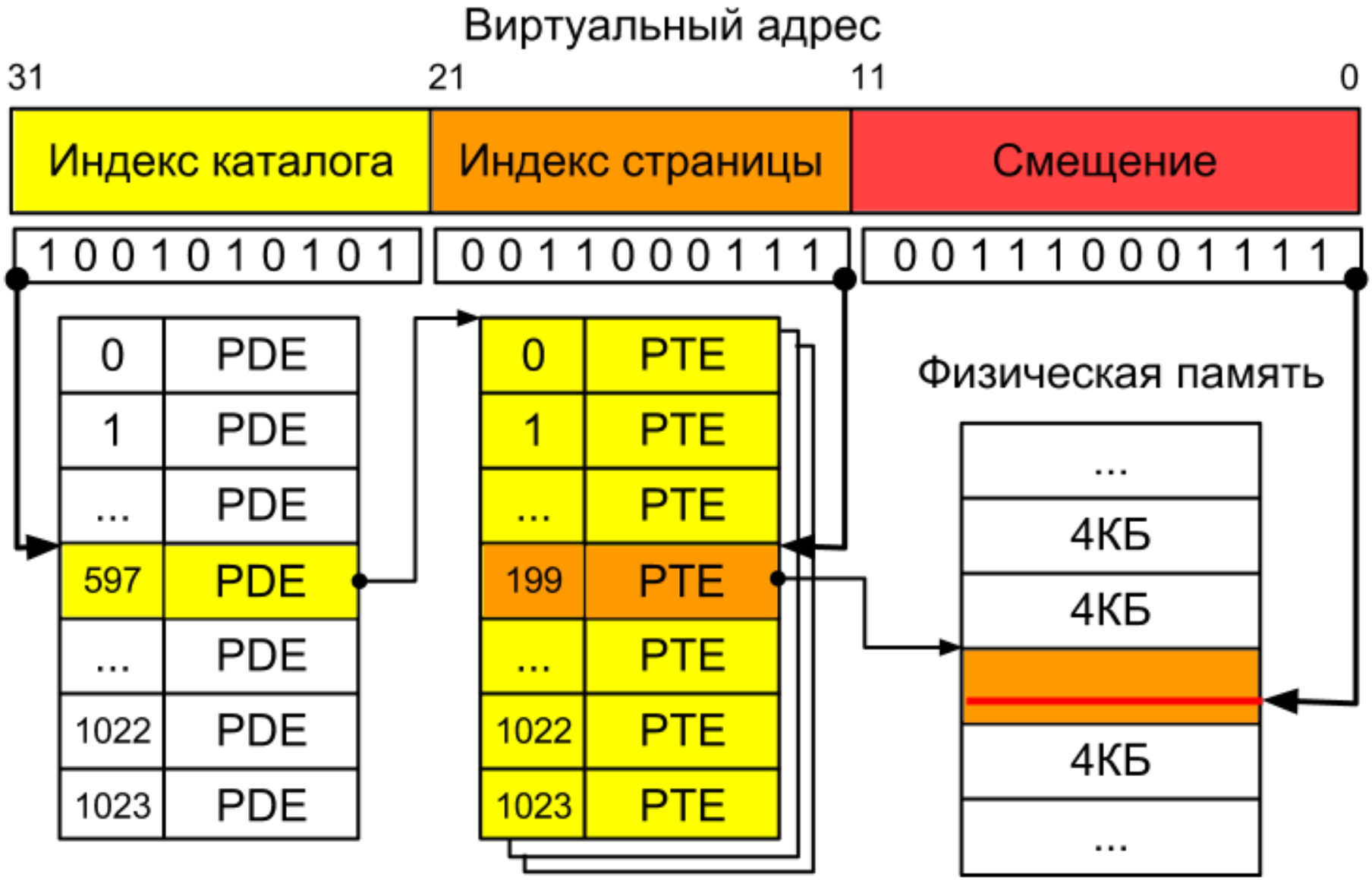
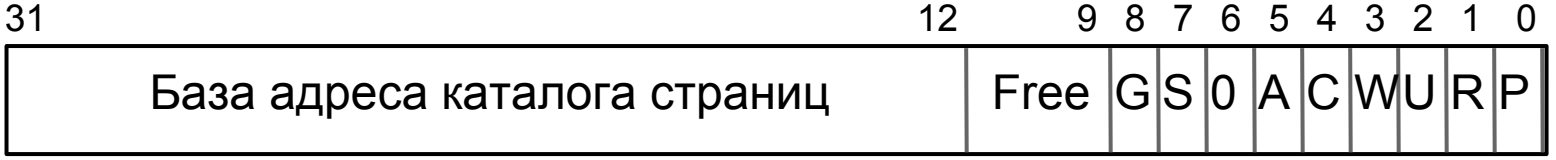


Схема страничной трансляции



Формат дескрипторов

PDE - запись в таблице директорий



Free - Доступны для программиста
G(Global) - Управление буфером TLB
S(Size) - Размер страниц (4КБ/ 4МБ)
A(Accessed) - Бит обращения к каталогу
C(Cache Disable)- Отключение кэширования
W(Write-through) - Запись в обход кэша
U(User) - Уровень доступа
R(Read/Write) - права (чтение/запись)
P(Present) - Бит присутствия страницы

PTE - запись в каталоге страниц

Free - Доступны для программиста
G(Global) - Управление буфером TLB
D(Dirty) - Страница изменена
A(Accessed) - Бит обращения к странице
C(Cache Disable)- Отключение кэширования
W(Write-through) - Запись в обход кэша
U(User) - Уровень доступа
R(Read/Write) - права (чтение/запись)
P(Present) - Бит присутствия страницы

Создание таблиц трансляции

- Выделение памяти
- Заполнение записей базовыми значениями
- Загрузка адресов таблиц в управляющий регистр (CR3)
- Трансляция один к одному

Включение страничной адресации

31-й бит регистра CR0 отвечает за разрешение страничного преобразования

```
22 void mmu_on(void) {
23
24     asm (
25         "mov %cr0, %eax\n"
26         "or  $0x80000000, %eax\n"
27         "mov %eax, %cr0"
28     );
29 }
```

Трансляция адресов

- Вызов с одного виртуального адреса разных функций

```
68 void  __attribute__((aligned(PAGE_SIZE))) function1(void) {
69     printf("\n\tInside the first function\n");
70 }
71
72 void __attribute__((aligned(PAGE_SIZE))) function2(void) {
73     printf("\n\tInside the second function\n");
74 }
```

```
104     printf("\nPaging starting...\n");
105
106     /* enabling paging */
107     mmu_on();
108
109     function1();
110
111     printf("\nMapping a new function to the old address\n");
112     map_region(function1,function2);
113
114     function1();
115
116     printf ("\nEnding mmu testing...\n");
117
118     /* disabling paging */
119     mmu_off();
```

Результат

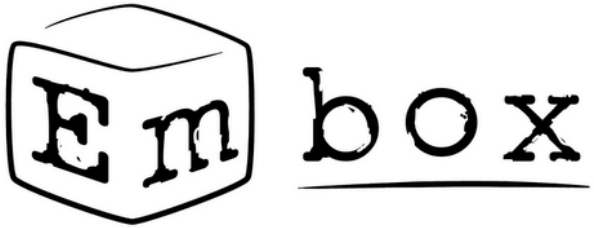
```
Welcome to Embox and have a lot of fun!
embox>mmuprobe

Paging starting...

        Inside the first function

Mapping a new function to the old address

        Inside the second function

embox>
```

Итог

- Реализован механизм страничной трансляции виртуальной памяти в рамках ОСРВ Embox

- Проверен механизм трансляции страниц

Контакты



Essential toolbox for embedded development

Embox Project Homepage

- **http://code.google.com/p/embox/**

Список рассылки

- **embox-sp@googlegroups.com**