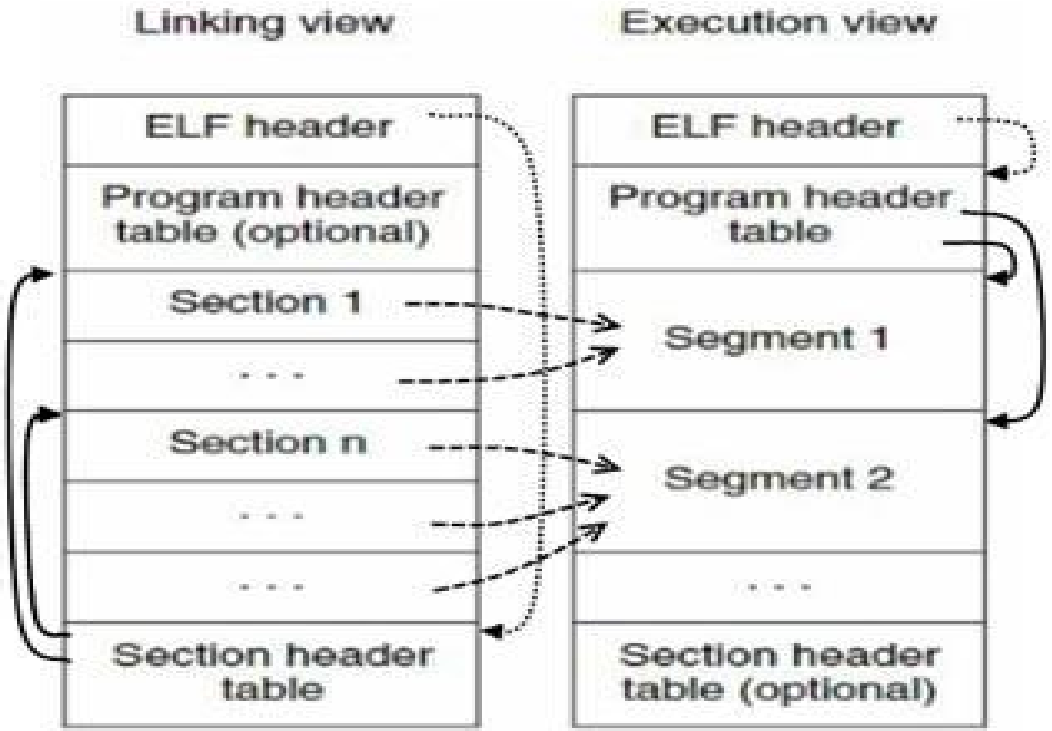
ОС

Компоновка и загрузка программ

Сборка программы

- **gcc *.c -o program**
  - обработка:
    - препроцессинг (gcc -E *.c -o *.c)
    - компиляция (gcc -S *.c -o *.s)
    - ассемблирование (gcc -c -o *.o)
    - компоновка/линковка (gcc *.o -o program)
  - выход - бинарный файл:
    - ELF
    - COFF
    - PE

ELF Format


Linking view
Execution view
ELF header
Program header table (optional)
Section 1
. . .
Section n
. . .
. . .
Section header table
ELF header
Program header table
Segment 1
Segment 2
. . .
Section header table (optional)

ELF Header

- **readelf -h <file>**
  - Magic - позволяет отличить elf файл от другого формата
  - Класс - LEF64, ELF32 и тд
  - Данные - little endian/big endian и тд
  - Тип - исполняемый, перемещаемый или разделяемый
  - Адрес точки входа - указатель на “main”
  - Начало заголовков программы
  - Начало заголовков секций

Таблица секций

- **readelf -S <file>**
  - Массив заголовков секций:
    - Имя секции
    - Тип секции - REL, SYMTAB, STRTAB и тд
    - Адрес - адрес в памяти, где будет расположена секция
    - Смещение - смещение секции от начала файла
    - Размер - размер секции

Таблицы символов и строк

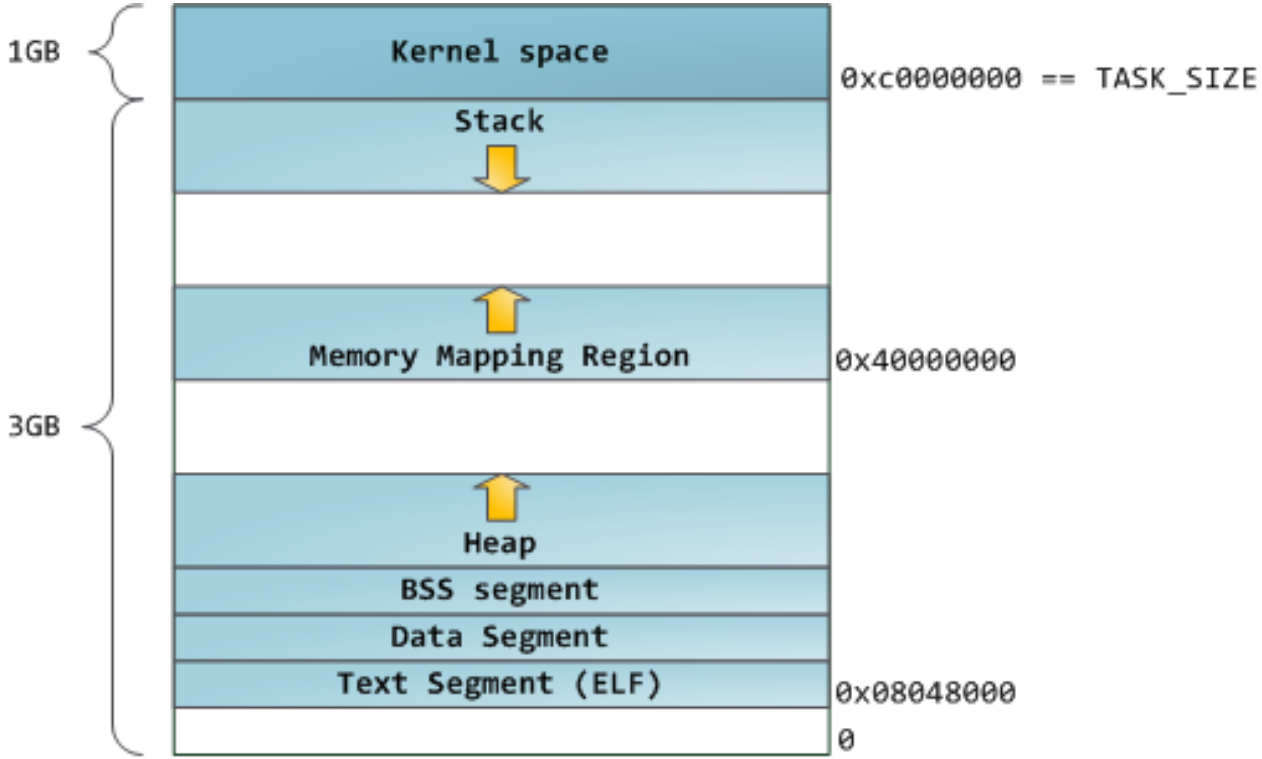
- **readelf -s <file>**
  - Имя - имя символа
  - Тип - тип символа (FILE, FUNC, SECTION, OBJECT)
  - Связывание - LOCAL или GLOBAL

- **readelf -p ‘.strtab’ <file>** - таблица строк

Основные секции ELF

- .data - содержит инициализированные данные
- .rodata - данные только для чтения (обычно строковые литералы)
- .bss - не инициализированные данные, выделяется в момент исполнения
- .text - содержит код

Процесс в памяти



Релокации

- Типичные имена:
  - .text.rel
  - .text.rela
  - .<section>.rel
  - .<section>.rela

- Каждая релокация содержит:
  - offset - смещение внутри секции
  - info - индекс в таблице символов + тип релокации

Компоновка/линковка

- ld - Link eDitor (редактор связей)
- Статическая:
  - компоновка *.o файлов
  - компоновка *.a архивов (с помощью утилиты ar)

- Динамическая:
  - связывание *.so - динамических библиотек (флаг -shared)

Заголовки программы

- **readelf -l <file>**
  - Массив заголовков программы:
    - Смещение секции в файле
    - Виртуальный адрес в памяти
    - Размер
    - Различные флаги (Read, Write, Execute)
    - Тип:
      - INTERP
      - LOAD
      - DYNAMIC

Покажи пример!

(hello)

Shared Objects

- Shared Object (*.so) - динамическая разделяемая библиотека (aka dll):
  - *.so хранится в памяти в единственном экземпляре
  - *.so используется одновременно несколькими процессами
  - *.so видна разным процессам под разными адресами

Динамическая линковка

- ld-linux.so - динамический компоновщик/загрузчик (ELF):
  - Ищет необходимые программе *.so библиотеки и загружает:
    - DT_RPATH (хранится в исполняемом файле, deprecated)
    - LD_LIBRARY_PATH (переменная окружения)
    - DT_RUNPATH (хранится в исполняемом файле)

- Как загрузить библиотеку самостоятельно:
  - dlopen()
  - dlsym()
  - dlclose()

Релокации

- .got - Global Offset Table:
  - хранит адреса динамических объектов

- .plt - Procedure Linkage Table:
  - набор “оберток”, которые используют секцию релокаций (.rela.plt), чтобы вызывать функции
  - Пример:

```
jmpq *<offset>(%rip)
```

Покажи пример!

(test_f и test_d)

Q&A

Другие форматы

- Common Object File Format (COFF)
- Mach-O
- a.out
- EXE (PE, LX, LE, NE, MZ)
- COM

Релокации

```
typedef struct {
    Elf32_Addr r_offset;
    Elf32_Word r_info;
    Elf32_Sword r_addend;
} Elf32_Rela;
```