OS, ELF

Carmi Merimovich

Tel-Aviv Academic College

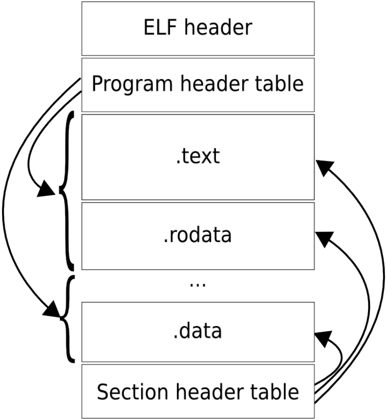
January 10, 2017

(static) Executlabe and Linkable Format (ELF)

ELF components

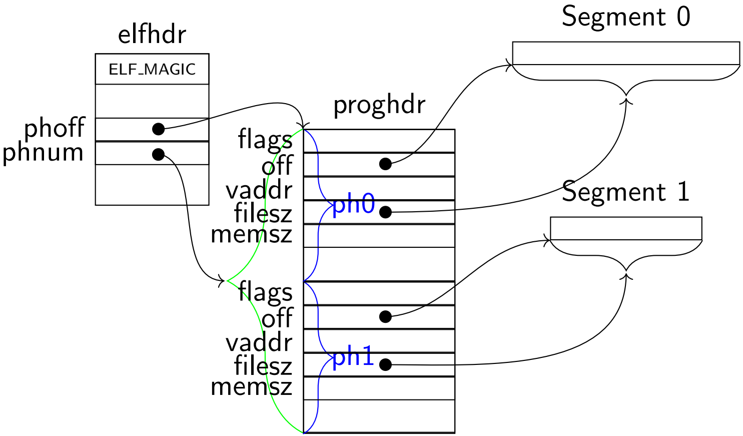
1. ELF header. Must begin at byte zero of the file.
2. PROGHDR vector.
3. Program segments.

ELF file, very abstract

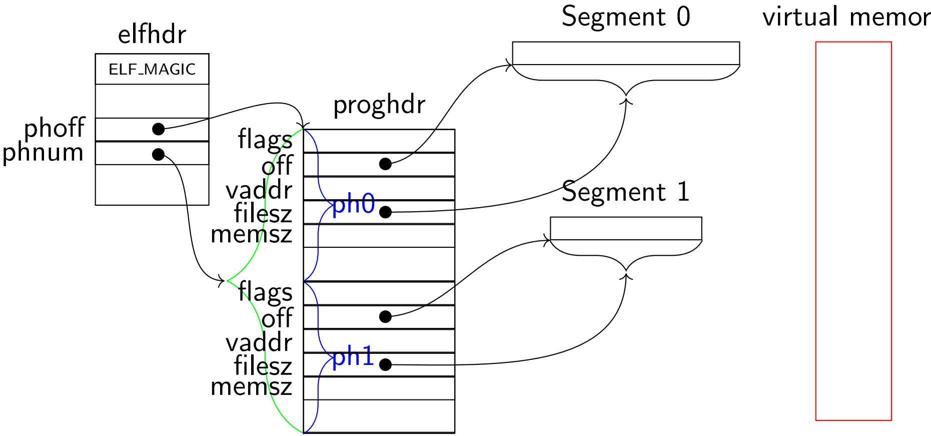


By Suruea - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=2922605

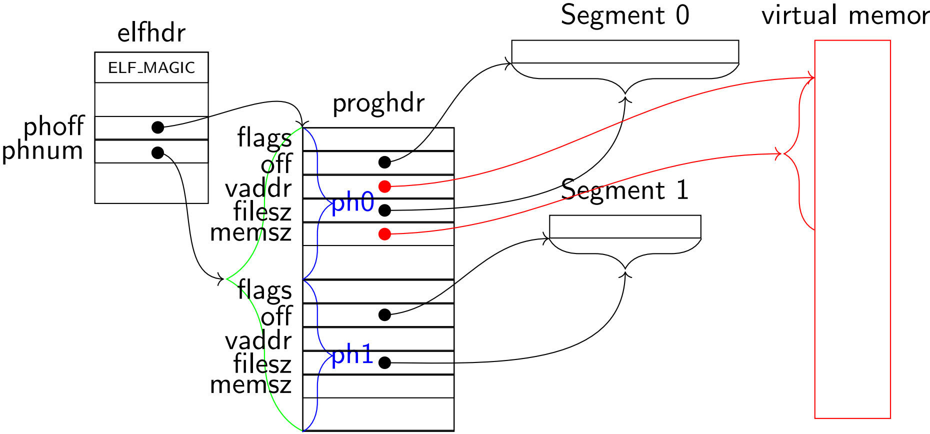
ELF, more detailed



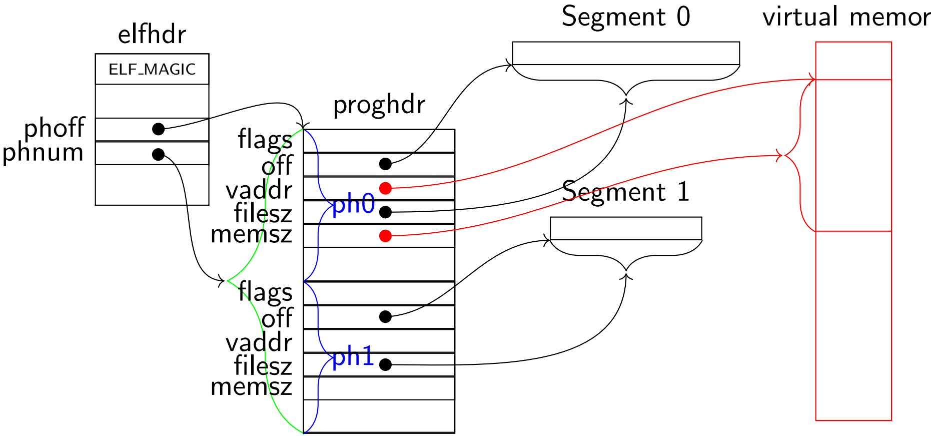
ELF, more detailed



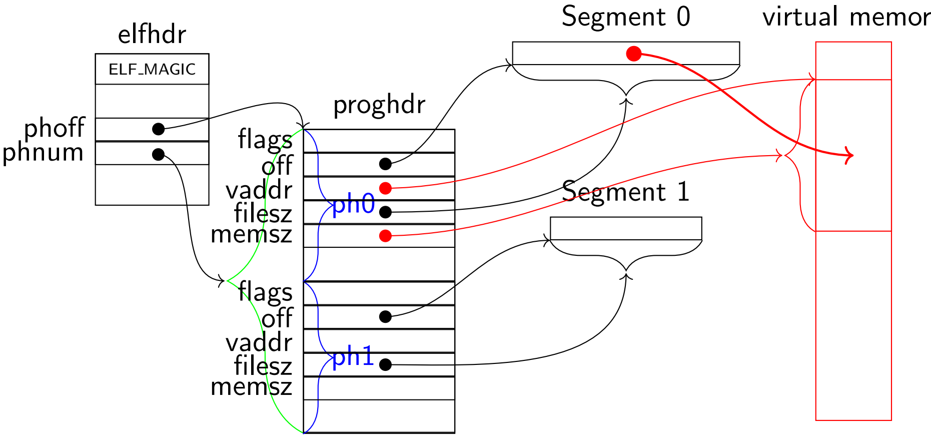
ELF, more detailed



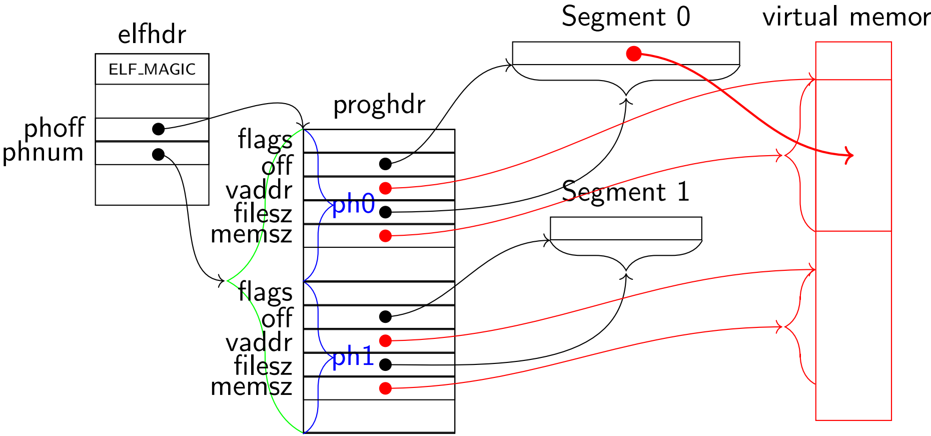
ELF, more detailed



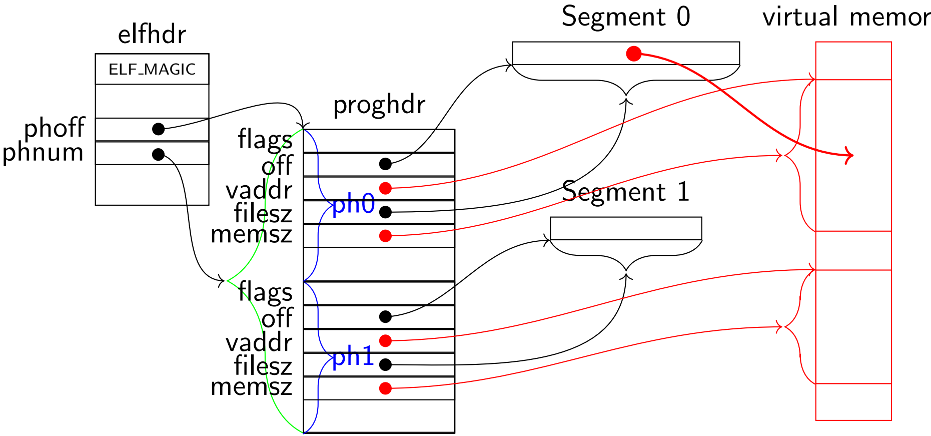
ELF, more detailed



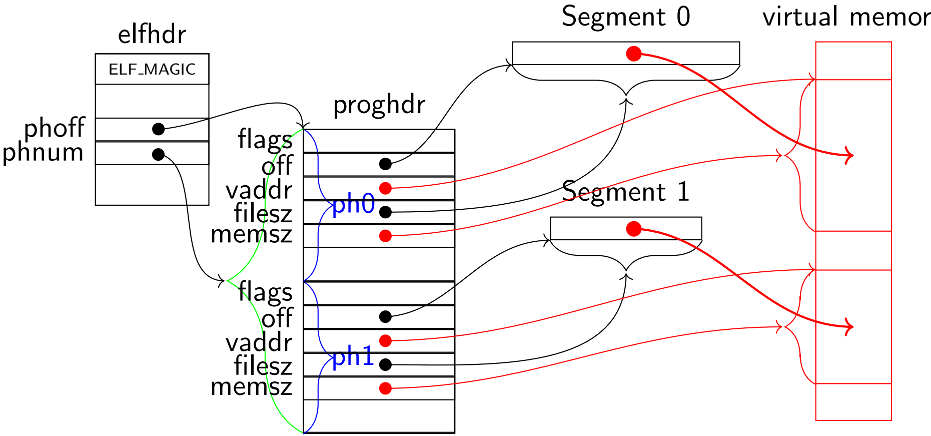
ELF, more detailed



ELF, more detailed



ELF, more detailed



ELFHDR

```
struct elfhdr {
  uint magic; // must equal ELF_MAGIC
  uchar elf[12];
  ushort type;
  ushort machine;
  uint version;
  uint entry;  // Entry point
  uint phoff;  // (File) Location of PROGHDR vectors
  uint shoff;
  uint flags;  // flag
  ushort ehsize;
  ushort phentsize;
  ushort phnum; // Length of PROGHDR vector
  ushort shentsize;
  ushort shnum;
  ushort shstrndx;
};
```

ELFHDR fields we are interested in

- **magic**: Should be ELF_MAGIC (0x464C457F).
- **entry**: Virtual address the program is starting at.
- **phoff**: File offset the Program segments Headers vector begins at.
- **phnum**: Number of elements in the Program segments Headers vector.

For each segment there is PROGHDR

```c
struct proghdr {
 uint type;   // Only PROG_LOAD matters to us
 uint off;    // Section location in file
 uint vaddr;  // Virtual address of section
 uint paddr;  // Physical address of section
 uint filesz;// Section size in file
 uint memsz;  // Section size in memory
 uint flags;
 uint align;
};
```