

Assignment 9

11-29-23

Kushagra Khatwani (kk5395)

Ans 1.

Variable Sized Partitioning:

- The physical memory is divided into variable sized partitions and regions, memory allocation is done dynamically at runtime.
- The addressing in variable sized partitioning is typically performed using absolute or relative addressing.

Segmentation:

- Segmentation divides the logical address space of a process into segments, where each segment represents specific type of data or code (eg. code segment, data segment).
- Segmentation uses a two level addressing mechanism with a segment register and an offset, allowing more flexible and logical organization of the memory.

Ans 2.

Paging:

- In paging, physical memory and logical address space are divided into fixed-size blocks known as pages and page frames, memory allocation occurs in fixed-size units and the size of pages is determined by the hardware and operating system and is typically a power of 2.
- Paging uses a simple, uniform addressing mechanism. Logical addresses consist of a page number and an offset within the page.

Segmentation:

- In segmentation, memory is divided into variable-sized segments, each representing a logical unit of a program, such as a code segment, data segment, or stack. The size of segments is not fixed and depends on the organization of the program.

- Segmentation uses a more complex two-level addressing mechanism. Logical addresses consist of a segment identifier and an offset within the segment.

Ans 3.

1.

```

kushagra@kushagra-virtual-machine: ~/intro_to_os/kk5395_hw9
kushagra@kushagra-virtual-machine: ~/intro_to_os/kk5395_hw9$ ./lab9_producer.o 10 7
Producer: Shared Buffer Start Address: 0x7fb4631f000
Producer: n Address: 0x5b9ffa010
kushagra@kushagra-virtual-machine: ~/intro_to_os/kk5395_hw9$

```

(a) Producer

```

kushagra@kushagra-virtual-machine: ~/intro_to_os/kk5395_hw9
kushagra@kushagra-virtual-machine: ~/intro_to_os/kk5395_hw9$ ./lab9_consumer.o 10 7
Consumer: Shared Buffer Start Address: 0x7fed8b6d000
Consumer: n Address: 0x55a4b9e5010
Consumer Printing...
1st element of the sequence is 0.
2nd element of the sequence is 7.
3rd element of the sequence is 14.
4th element of the sequence is 21.
5th element of the sequence is 28.
6th element of the sequence is 35.
7th element of the sequence is 42.
8th element of the sequence is 49.
9th element of the sequence is 56.
10th element of the sequence is 63.
kushagra@kushagra-virtual-machine: ~/intro_to_os/kk5395_hw9$

```

(b) Consumer

Figure 1: Address of buffer shown in images for consumer and producer

i) The address printed is virtual address. The operating system provides each process with a virtual address space that is mapped to physical memory.

2.

```

kushagra@kushagra-virtual-machine: ~/intro_to_os/kk5395_hw9
kushagra@kushagra-virtual-machine: ~/intro_to_os/kk5395_hw9$ ./lab9_producer.o 10 7
Producer: Shared Buffer Start Address: 0x7fb4631f000
Producer: n Address: 0x55b6ff4fa010
kushagra@kushagra-virtual-machine: ~/intro_to_os/kk5395_hw9$

```

(a) Producer Running Program

```

kushagra@kushagra-virtual-machine: ~/intro_to_os/kk5395_hw9$ readelf -s lab9_producer.o | grep n
Symbol table '.dynsym' contains 18 entries:
Num: Value Size Type Bind Vis Ndx Name
14: 0000000000000000 0 NOTYPE WEAK DEFAULT UND _glibc_start
Symbol table '.symtab' contains 52 entries:
Num: Value Size Type Bind Vis Ndx Name
4: 0000000000000110 0 FUNC LOCAL DEFAULT 16 deregister_tm_cloes
5: 0000000000000120 0 FUNC LOCAL DEFAULT 16 register_tm_cloes
10: 0000000000000160 0 OBJECT LOCAL DEFAULT 21 _frame_dummy[...]
21: 000000000000012a 89 FUNC GLOBAL DEFAULT 16 insert_tobuffer
21: 00000000000004010 4 OBJECT GLOBAL DEFAULT 20 n
24: 0000000000000199 37 FUNC GLOBAL DEFAULT 16 _initializeBuffer
26: 0000000000000158 0 FUNC GLOBAL HIDDEN 17 _ftl
28: 0000000000000000 0 FUNC GLOBAL DEFAULT UND printf@GLIBC_2.2.5
29: 0000000000000000 0 FUNC GLOBAL DEFAULT UND fprintf@GLIBC_2.2.5
32: 0000000000000000 0 FUNC GLOBAL DEFAULT UND fprintf@GLIBC_2.2.5
33: 0000000000000000 0 NOTYPE WEAK DEFAULT UND _start
34: 0000000000000400 0 OBJECT GLOBAL HIDDEN 25 _do_handle
35: 0000000000000200 4 OBJECT GLOBAL DEFAULT 18 _to_stdin_used
36: 0000000000000430 0 NOTYPE GLOBAL DEFAULT 26 _id
39: 0000000000000000 0 FUNC GLOBAL DEFAULT UND mmap@GLIBC_2.2.5
40: 0000000000000135f 554 FUNC GLOBAL DEFAULT 16 _nati
45: 0000000000000000 0 FUNC GLOBAL DEFAULT UND shm_open@GLIBC_2.34
47: 0000000000000000 0 WEAK DEFAULT UND _cxa_finalize[...]
48: 0000000000000100 0 FUNC GLOBAL HIDDEN 12 _litt
49: 0000000000000000 0 FUNC GLOBAL DEFAULT UND _dereg@GLIBC_2.2.5
kushagra@kushagra-virtual-machine: ~/intro_to_os/kk5395_hw9$

```

(b) ELF file for Producer

Figure 2: Address of n in elf file and running program for producer.

i) The address for n is 0x55b6ff4fa010 in running program whereas in the ELF file it has a address 0000000000004010. Both the addresses are different because in the ELF file the address represents the virtual addresses at the time of compilation or linking whereas when we print the address from the running program it is different because OS can assign different base address each time when the program runs.

3.

```

kushagra@kushagra-virtual-machine: ~/intro_to_os/kk5395_hw9
kushagra@kushagra-virtual-machine: ~/intro_to_os/kk5395_hw9$ readelf --file-header lab9_producer.o

ELF Header:
  Magic:   7f 45 4c 46 02 01 01 00 00 00 00 00 00 00 00 00
  Class:   ELF64
  Data:    2's complement, little endian
  Version: 1 (current)
  OS/ABI:  UNIX - System V
  ABI Version:   0
  Type:    DYN (Position-Independent Executable file)
  Machine: Advanced Micro Devices X86-64
  Version:  0x1

  Entry point address: 0x11a0
  Start of program headers: 64 (bytes into file)
  Start of section headers: 14656 (bytes into file)
  Flags: 0x0
  Size of this header: 64 (bytes)
  Size of program headers: 56 (bytes)
  Number of program headers: 13
  Size of section headers: 64 (bytes)
  Number of section headers: 31
  Section header string table index: 30
kushagra@kushagra-virtual-machine: ~/intro_to_os/kk5395_hw9$

```

(a) Entry Point for Producer

```

kushagra@kushagra-virtual-machine: ~/intro_to_os/kk5395_hw9
kushagra@kushagra-virtual-machine: ~/intro_to_os/kk5395_hw9$ readelf --file-header lab9_consumer.o

ELF Header:
  Magic:   7f 45 4c 46 02 01 01 00 00 00 00 00 00 00 00 00
  Class:   ELF64
  Data:    2's complement, little endian
  Version: 1 (current)
  OS/ABI:  UNIX - System V
  ABI Version:   0
  Type:    DYN (Position-Independent Executable file)
  Machine: Advanced Micro Devices X86-64
  Version:  0x1

  Entry point address: 0x11a0
  Start of program headers: 64 (bytes into file)
  Start of section headers: 14656 (bytes into file)
  Flags: 0x0
  Size of this header: 64 (bytes)
  Size of program headers: 56 (bytes)
  Number of program headers: 13
  Size of section headers: 64 (bytes)
  Number of section headers: 31
  Section header string table index: 30
kushagra@kushagra-virtual-machine: ~/intro_to_os/kk5395_hw9$

```

(b) Entry Point for Consumer

Figure 3: Entry Point for Producer and Consumer

The entry point virtual address in both my producer and consumer was same, 0x11a0.