List of Figures

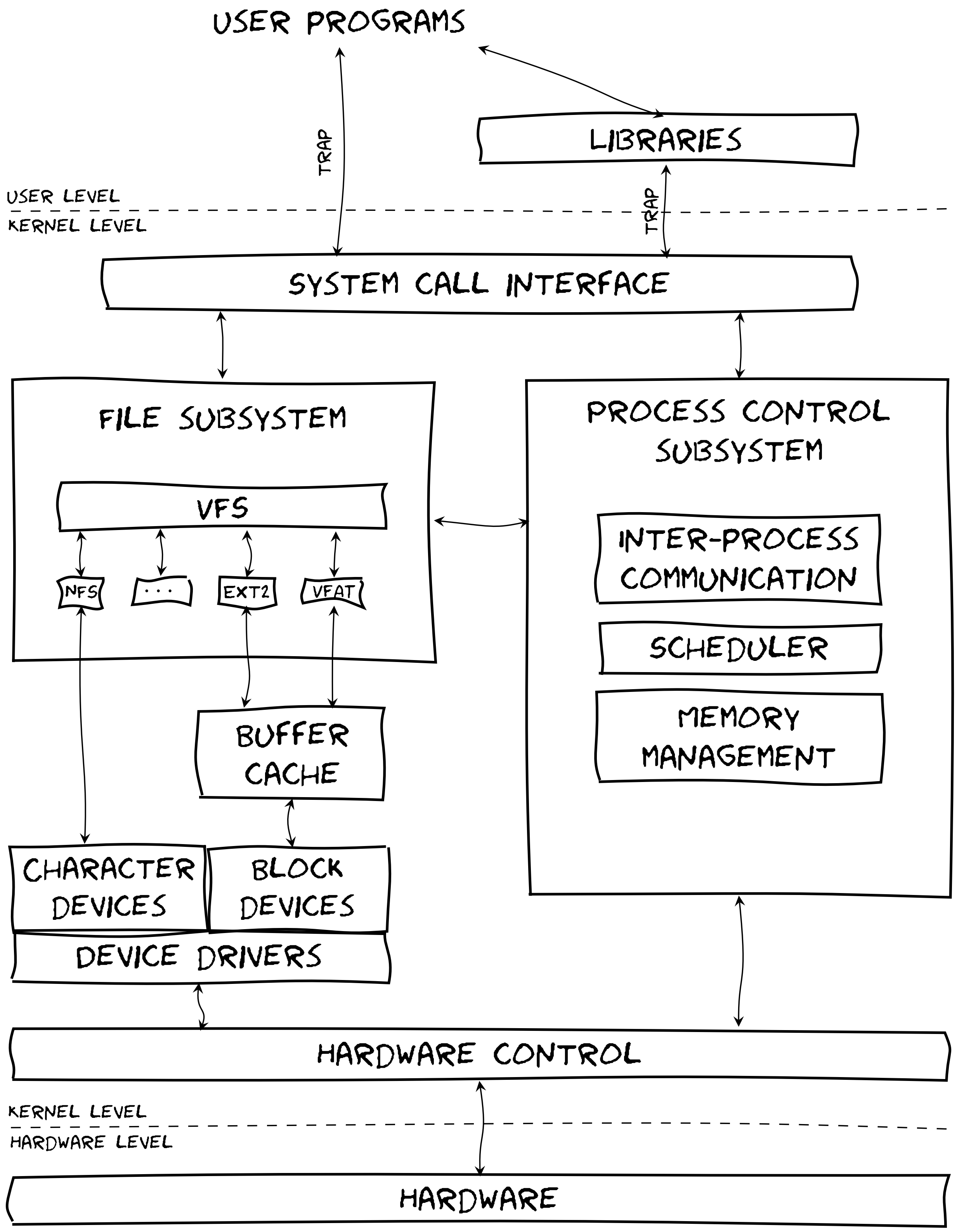


Fig. 1: OS overview

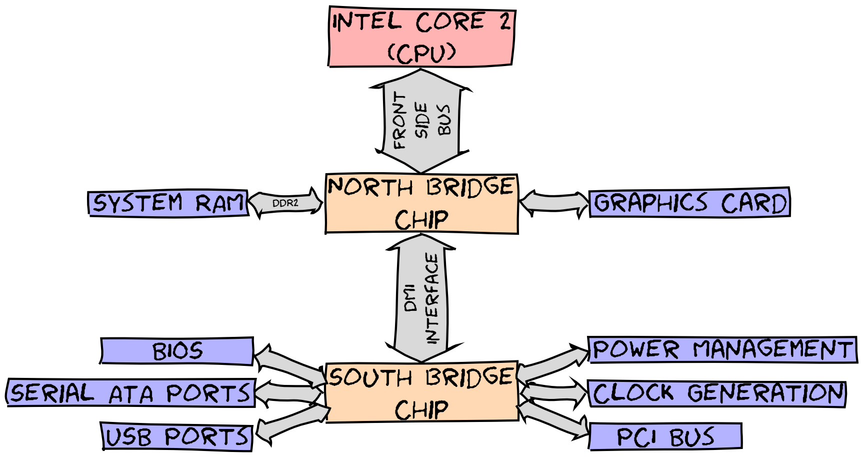


Fig. 2: Motherboard chipsets

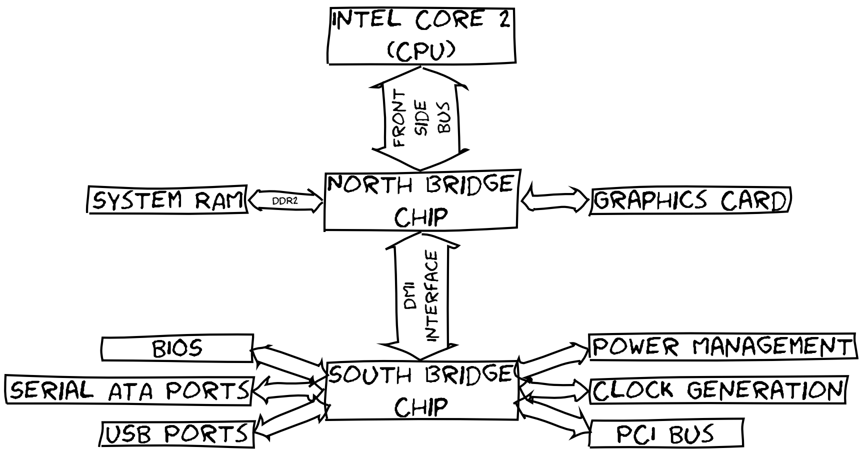


Fig. 3: Motherboard chipsets (bw version)

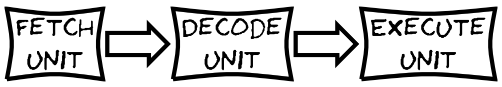


Fig. 4: CPU's working cycle

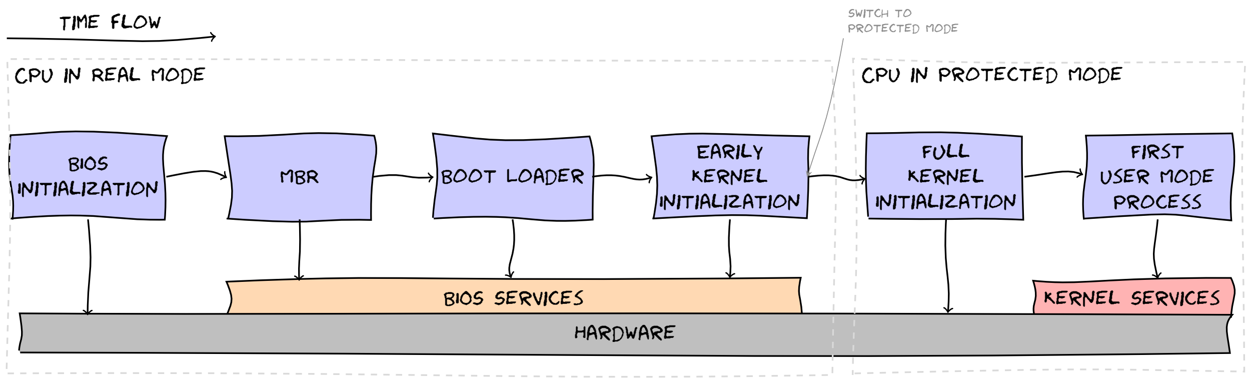


Fig. 5: Bootstrapping

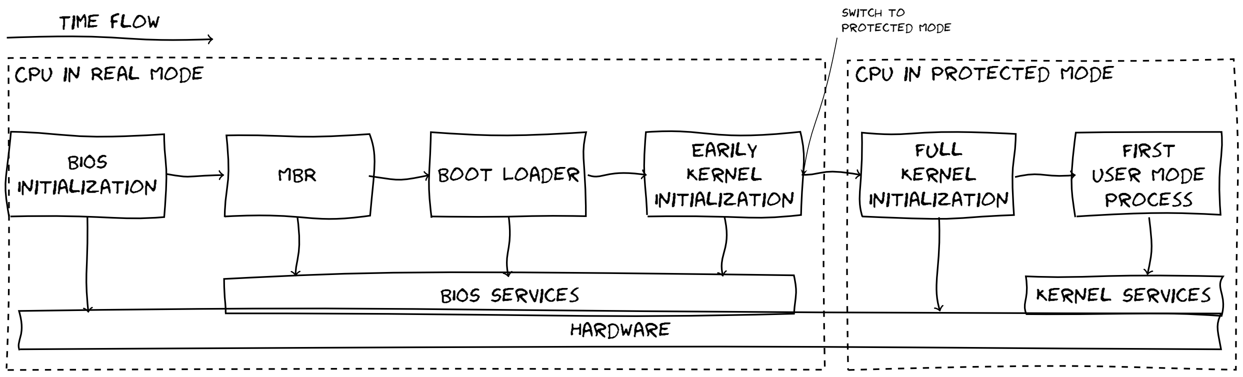


Fig. 6: Bootstrapping (bw version)

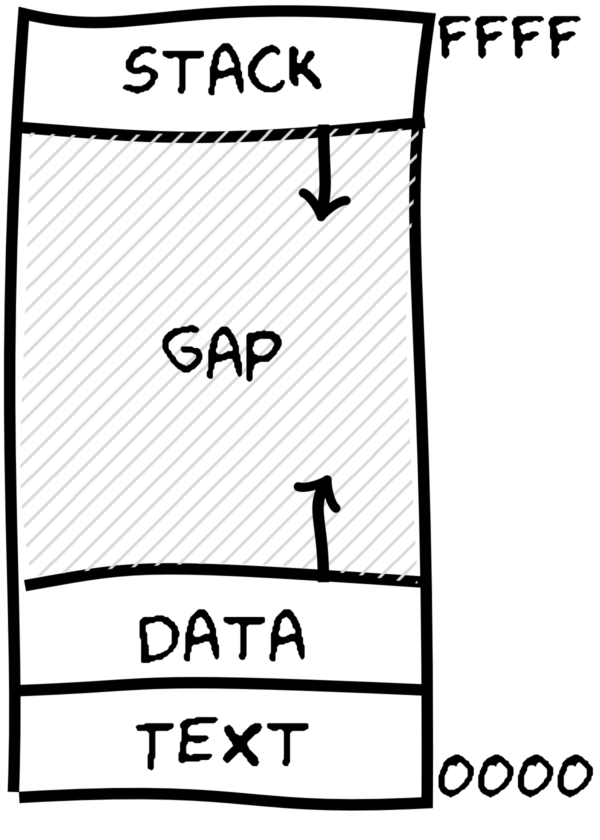


Fig. 7: Process’ virtual address space

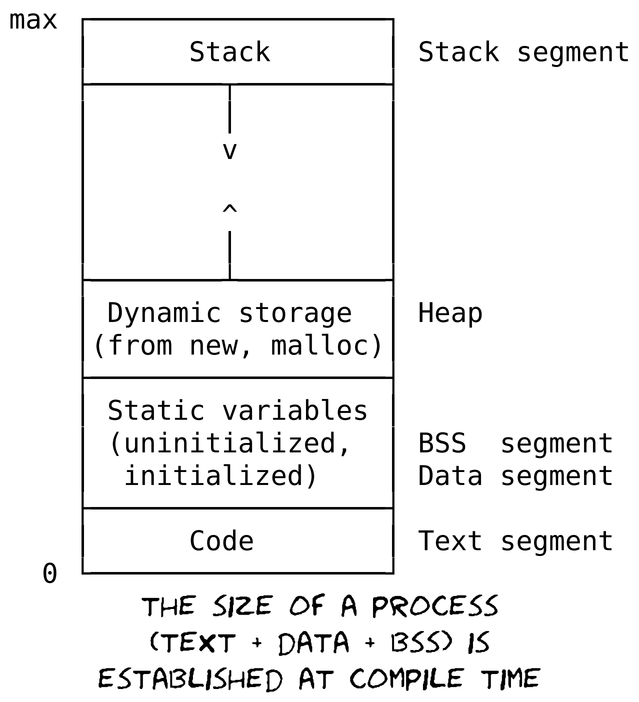

max
Stack
Stack segment
v
^
Dynamic storage (from new, malloc)
Heap
Static variables (uninitialized, initialized)
BSS segment
Data segment
Code
Text segment
0
THE SIZE OF A PROCESS (TEXT + DATA + BSS) IS ESTABLISHED AT COMPILE TIME

Fig. 8: UNIX view of a process

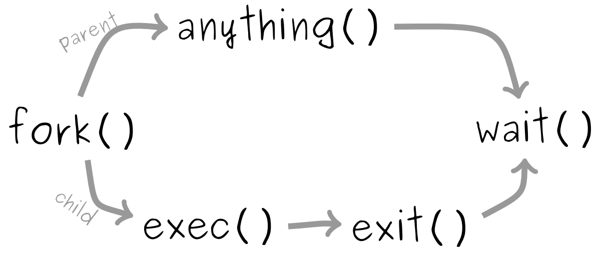


Fig. 9: Process creation

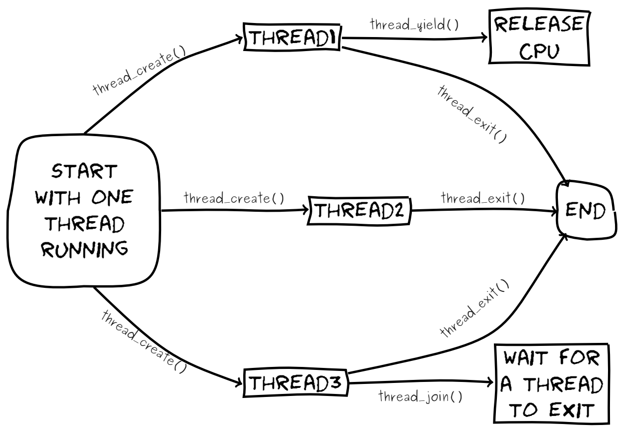


Fig. 10: Thread operations

```
typedef int semaphore;
    semaphore resource_1;
    semaphore resource_2;

    void process_A(void) {
        down(&resource_1);
        down(&resource_2);
        use_both_resources( );
        up(&resource_2);
        up(&resource_1);
    }

    void process_B(void) {
        down(&resource_1);
        down(&resource_2);
        use_both_resources( );
        up(&resource_2);
        up(&resource_1);
    }
```

(a)

```
    semaphore resource_1;
    semaphore resource_2;

    void process_A(void) {
        down(&resource_1);
        down(&resource_2);
        use_both_resources( );
        up(&resource_2);
        up(&resource_1);
    }

    void process_B(void) {
        down(&resource_2);
        down(&resource_1);
        use_both_resources( );
        up(&resource_1);
        up(&resource_2);
    }
```

Deadlock!

(b)

Fig. 11: Deadlock — Resource issues

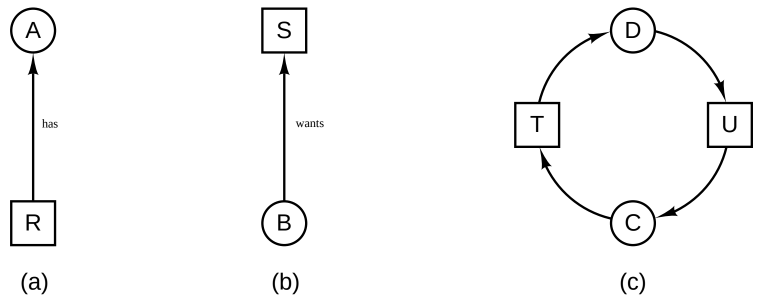


Fig. 12: Deadlock notions

	Has	Max
A	0	6
B	0	5
C	0	4
D	0	7

Free: 10

(a)

	Has	Max
A	1	6
B	1	5
C	2	4
D	4	7

Free: 2

(b)

	Has	Max
A	1	6
B	2	5
C	2	4
D	4	7

UNSAFE!

Free: 1

(c)

Fig. 13: Deadlock — Banker algorithm

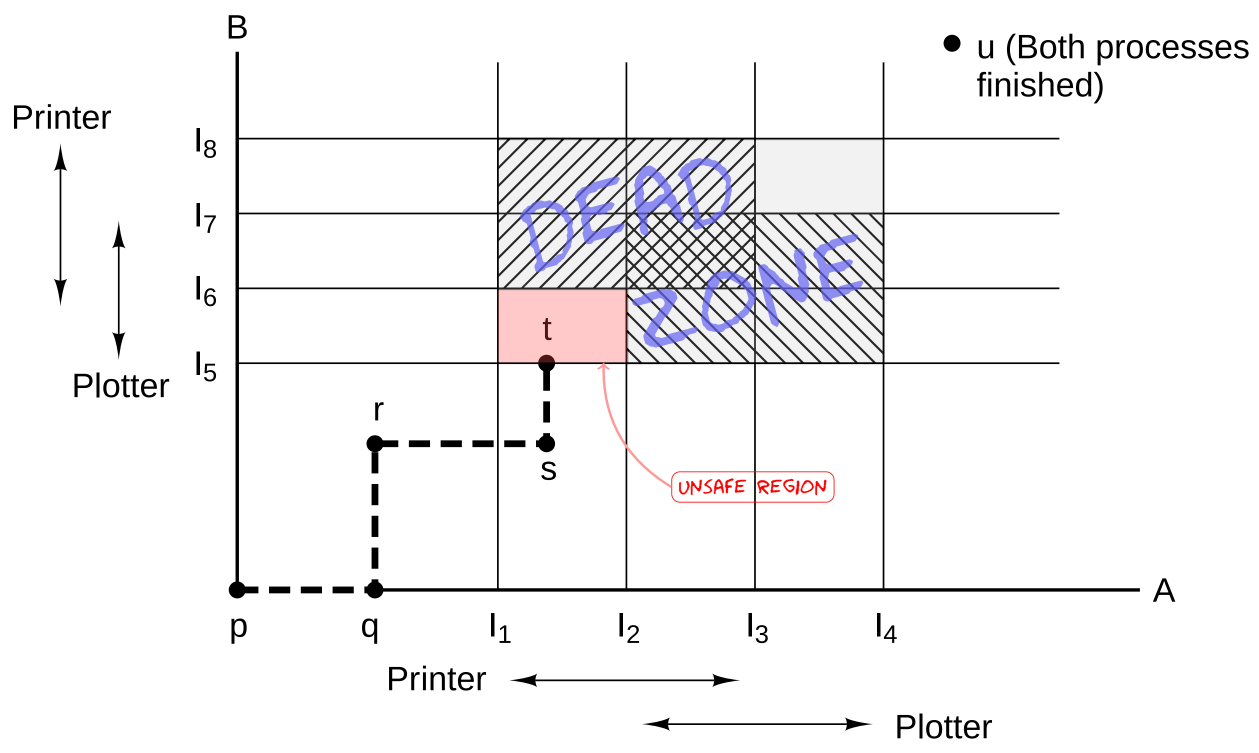


Fig. 14: Deadlock avoidance

	Has	Max
A	3	9
B	2	4
C	2	7

Free: 3
(a)

	Has	Max
A	4	9
B	2	4
C	2	7

UNSAFE

Free: 2
(b)

	Has	Max
A	4	9
B	4	4
C	2	7

Free: 0
(c)

	Has	Max
A	4	9
B	—	—
C	2	7

Free: 4
(d)

Fig. 15: Deadlock avoidance

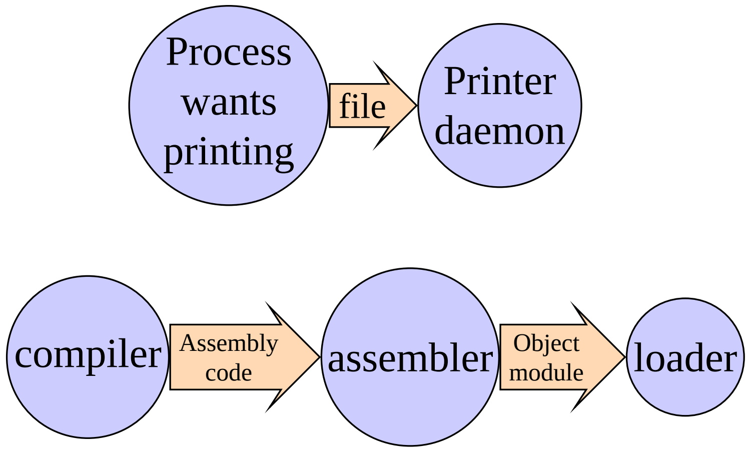


Fig. 16: Producers and consumers

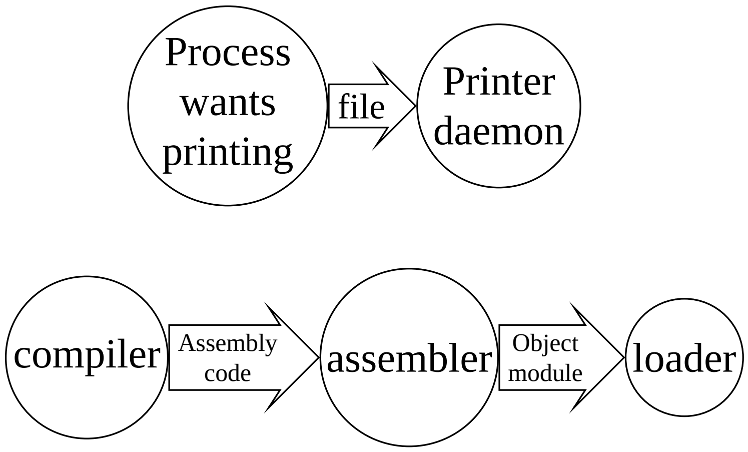


Fig. 17: Producers and consumers (bw version)

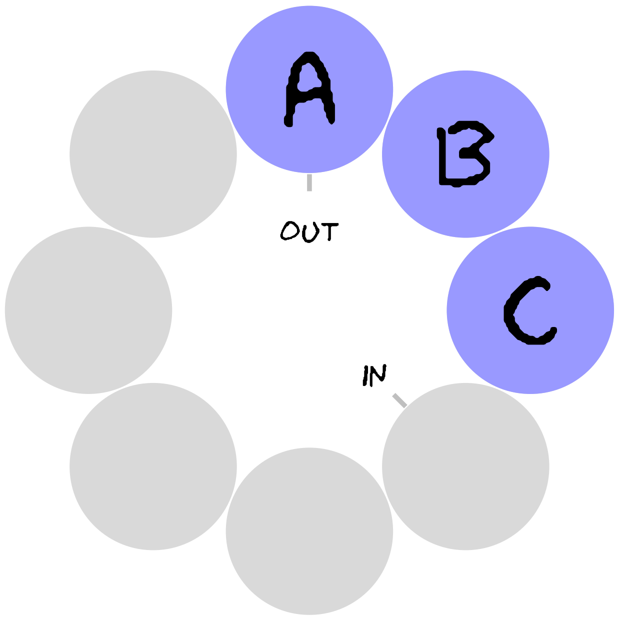


Fig. 18: A circular array

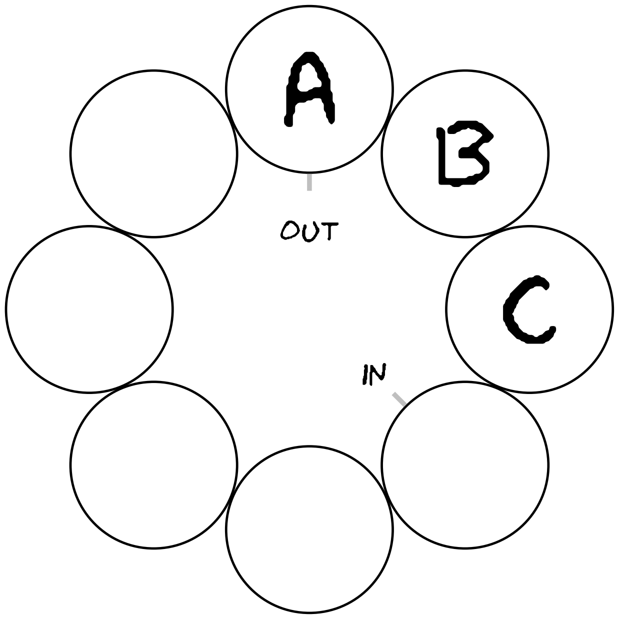


Fig. 19: A circular array (bw version)

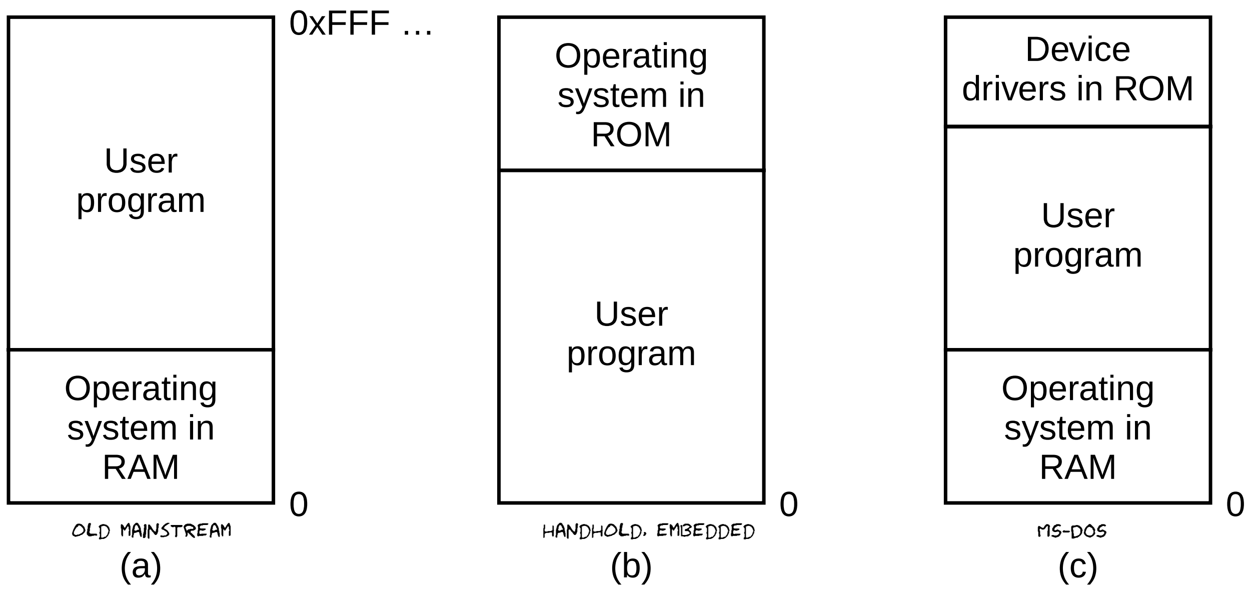


Fig. 20: Real mode memory layouts

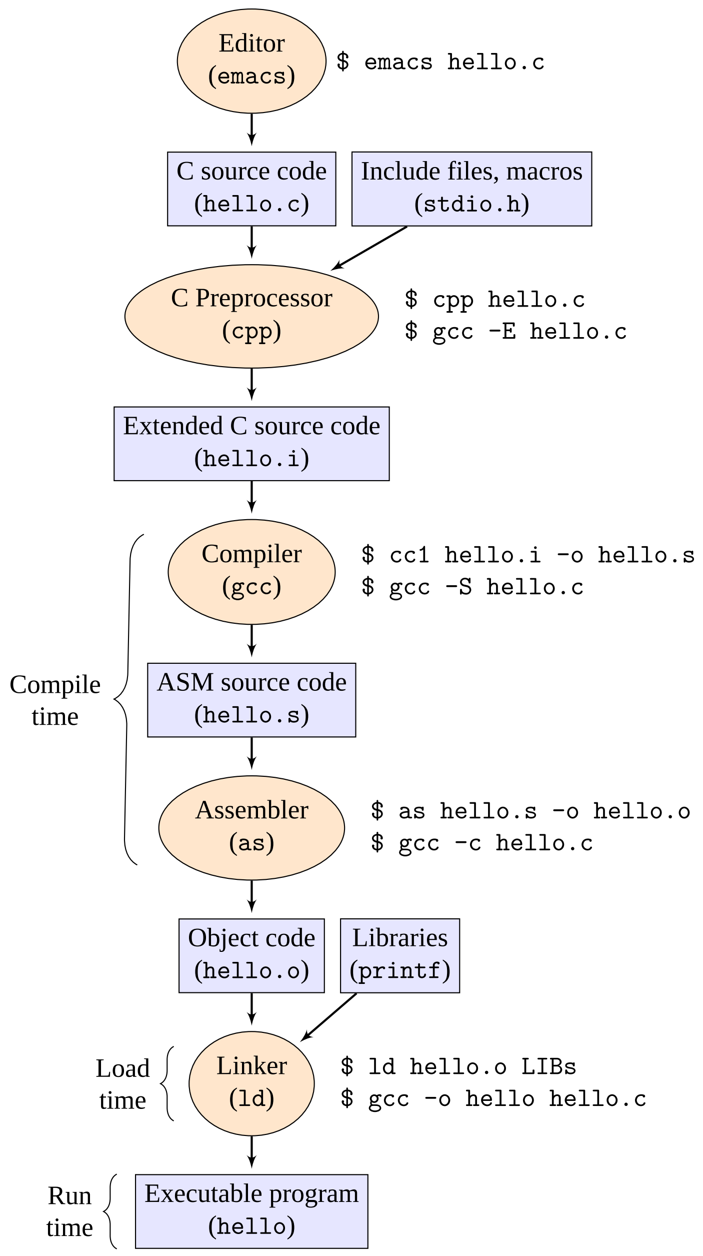


Fig. 21: Tool chain

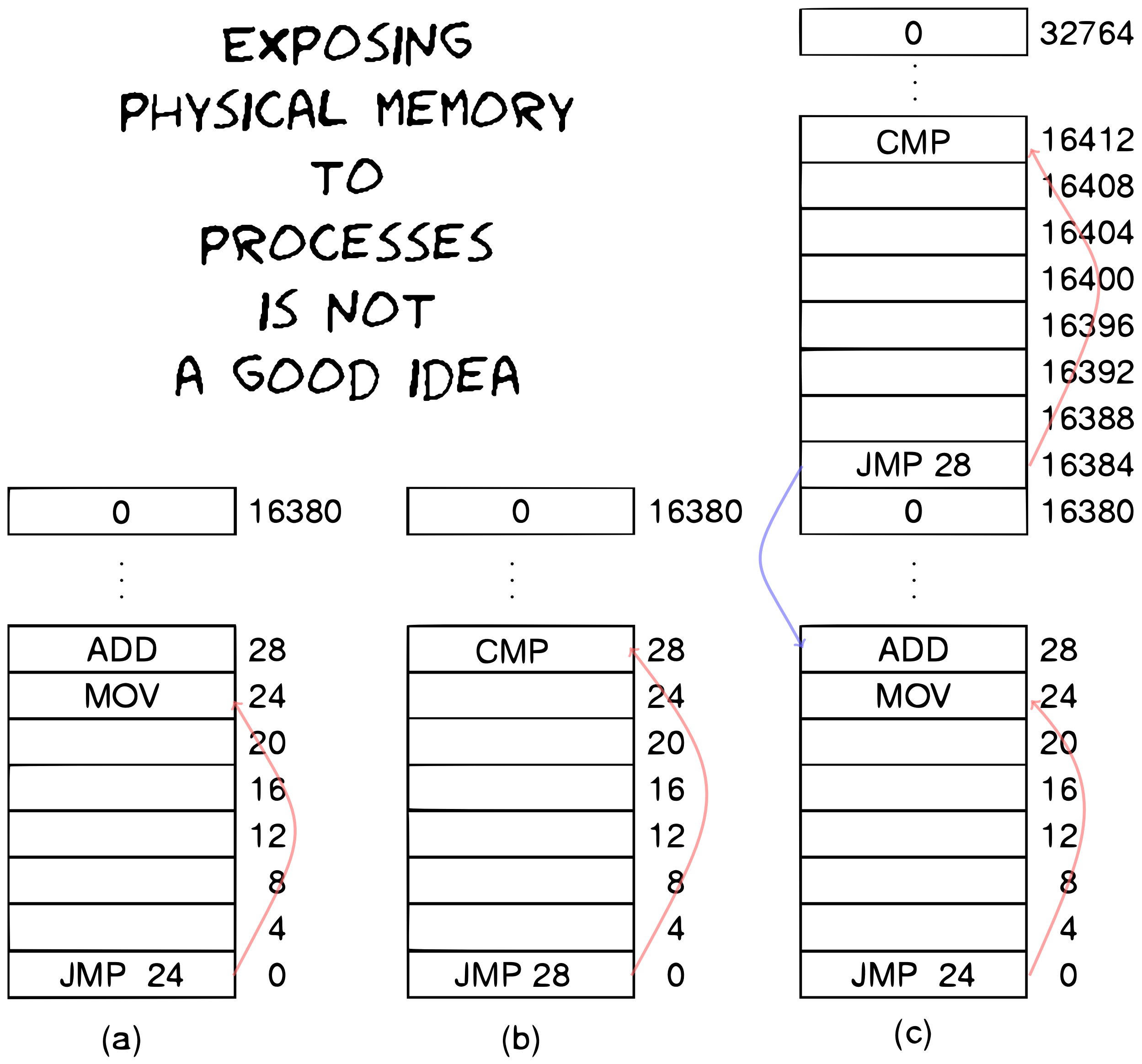


Fig. 22: Relocation

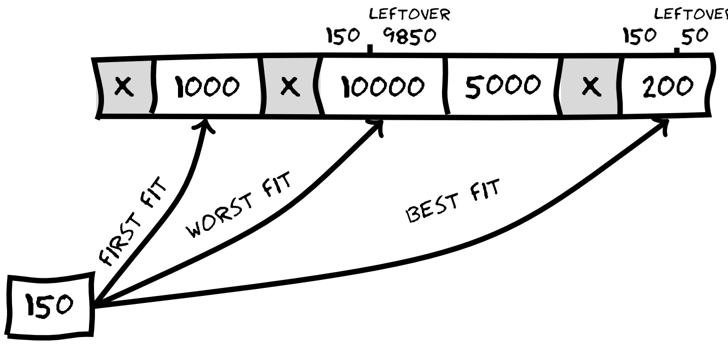


Fig. 23: First fit, best fit, worst fit

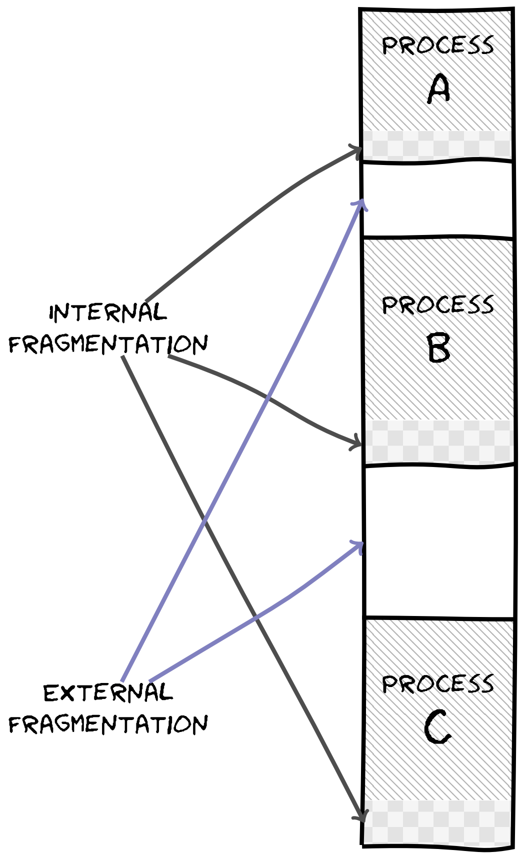


Fig. 24: Memory fragmentation

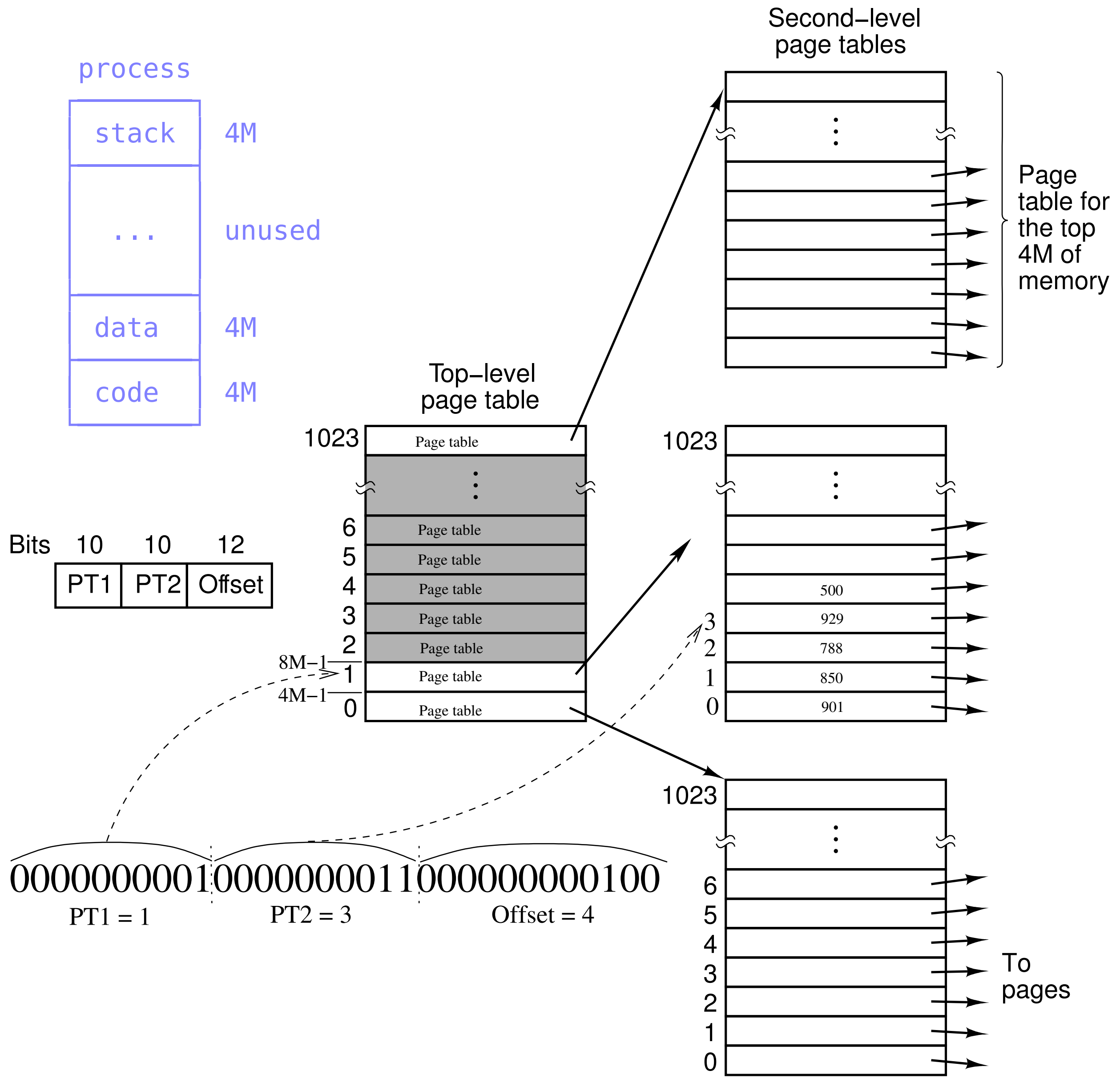


Fig. 25: Two-level paging

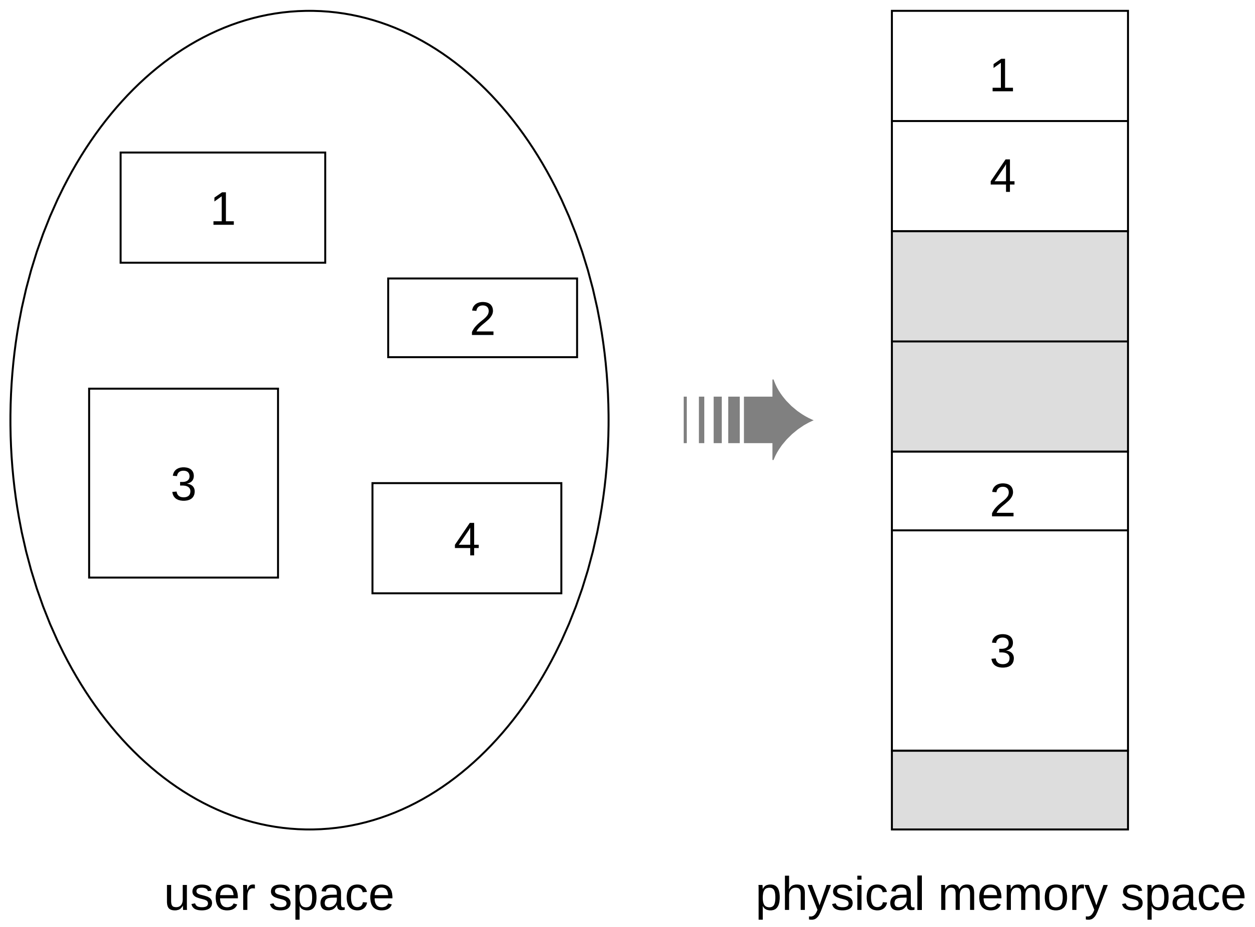
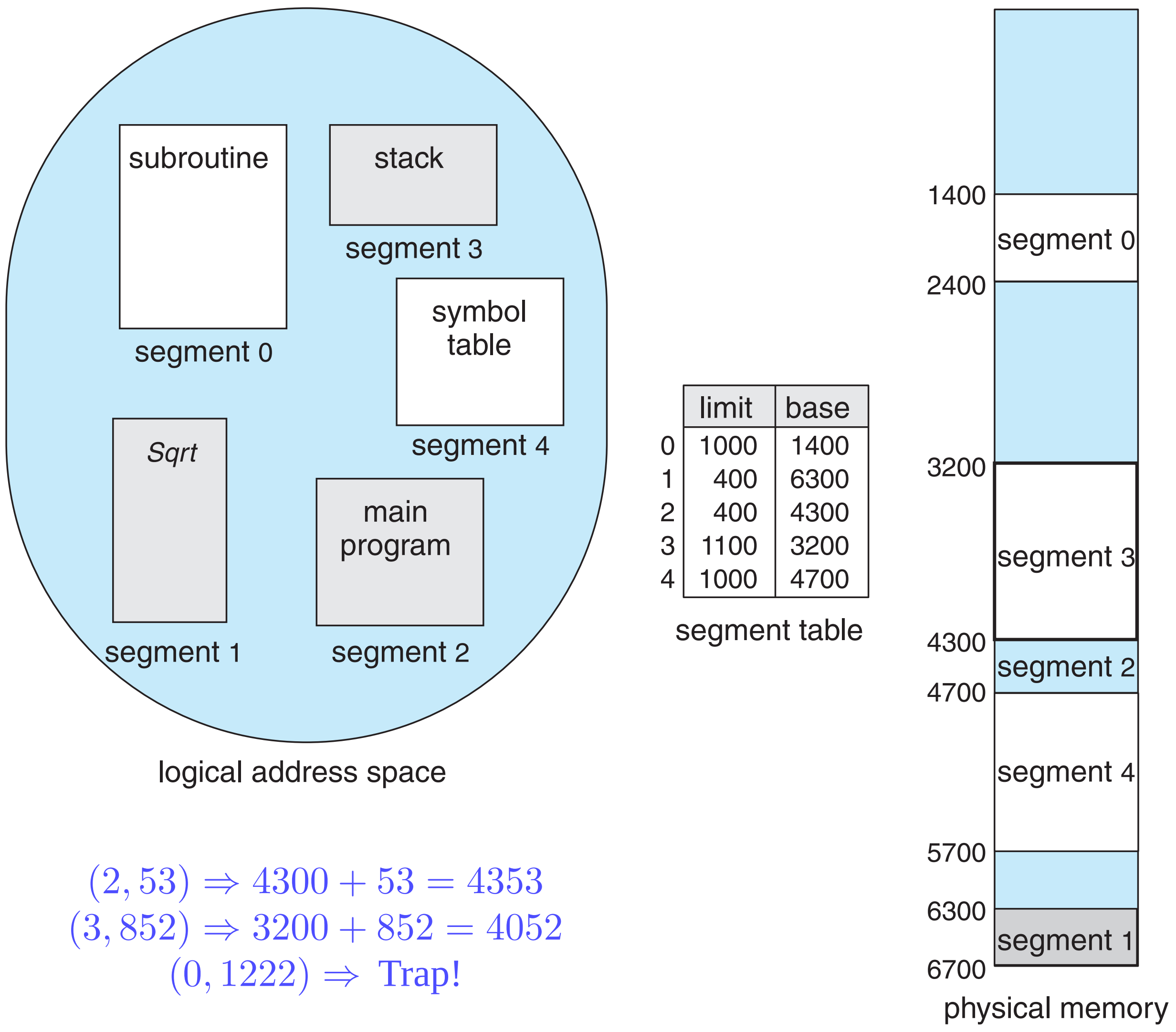


Fig. 26: Memory segmentation



	limit	base
0	1000	1400
1	400	6300
2	400	4300
3	1100	3200
4	1000	4700

$$(2,53) \Rightarrow 4300 + 53 = 4353$$
$$(3,852) \Rightarrow 3200 + 852 = 4052$$
$$(0,1222) \Rightarrow \text{ Trap!}$$

Fig. 27: Memory segmentation — Address translation

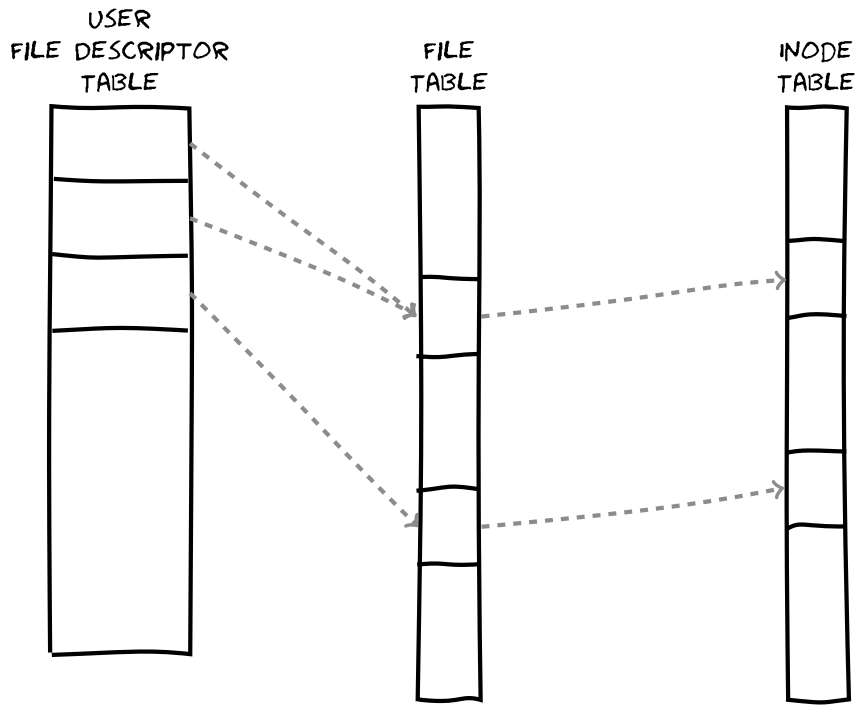


Fig. 28: File system tables

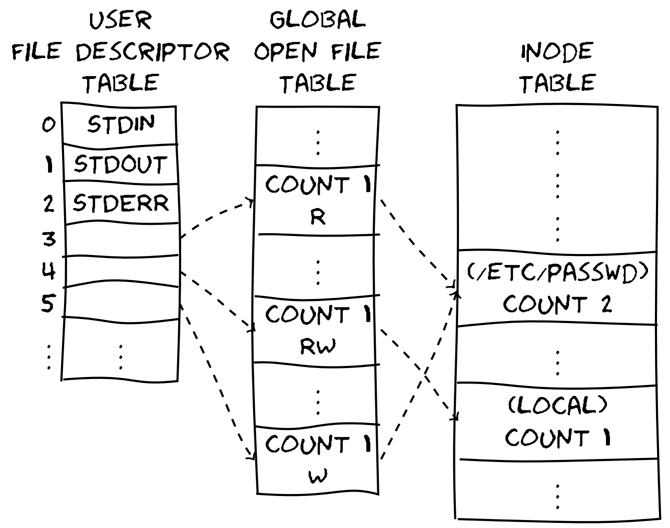


Fig. 29: File tables

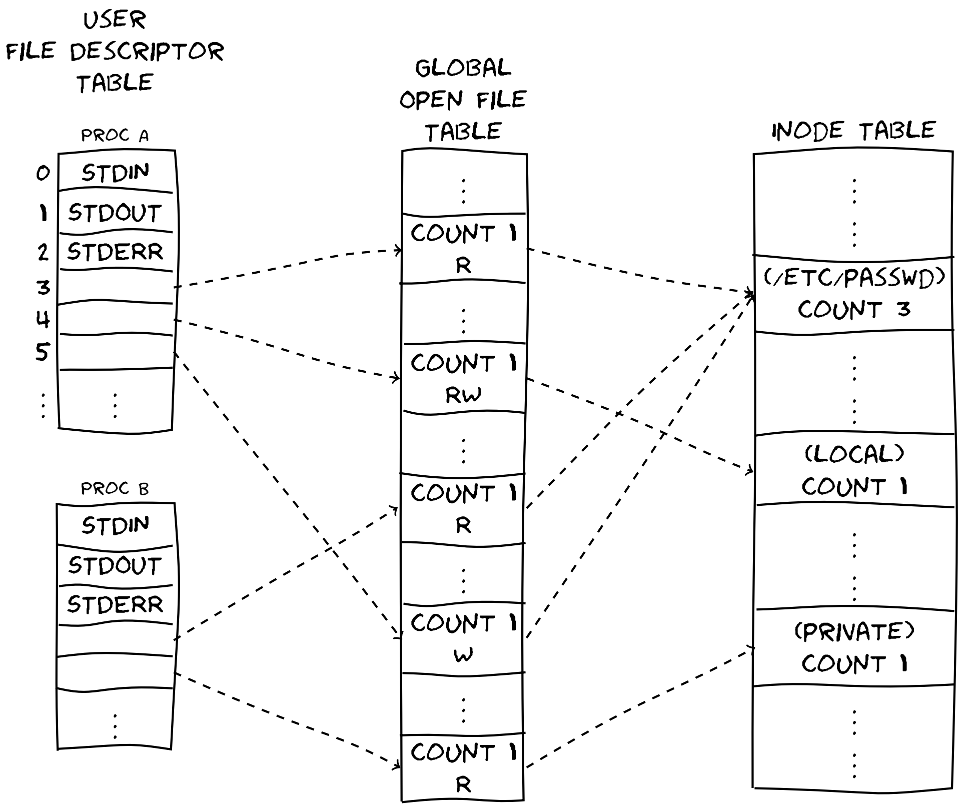


Fig. 30: File tables

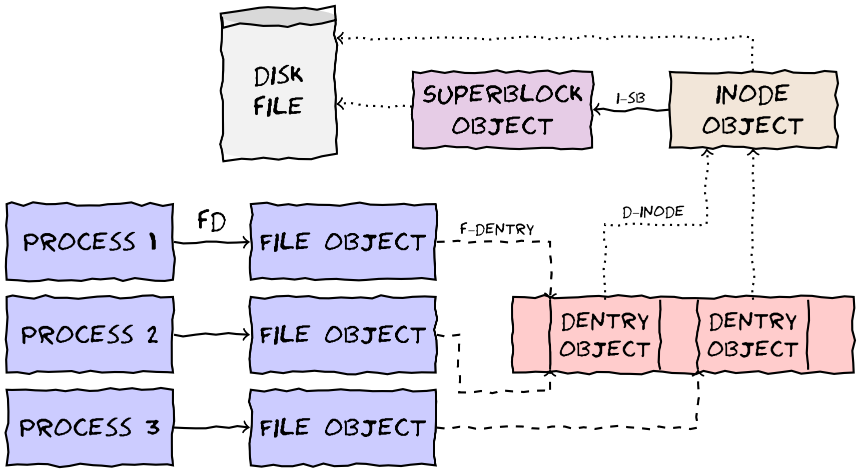


Fig. 31: VFS objects

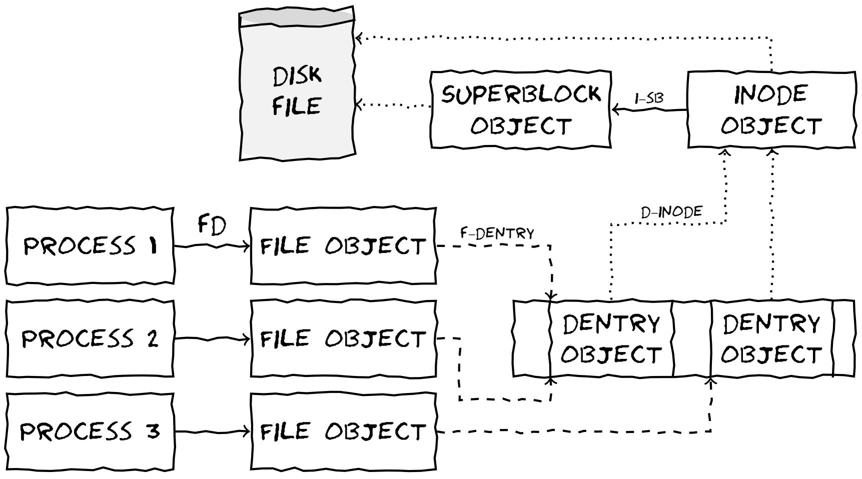


Fig. 32: VFS objects (bw version)

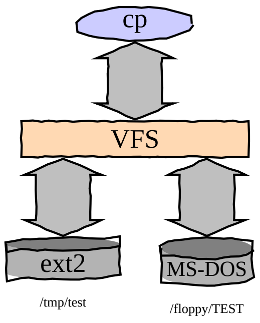


Fig. 33: VFS file copy

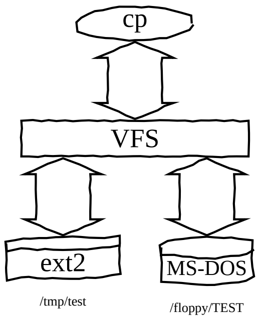


Fig. 34: VFS file copy (bw version)

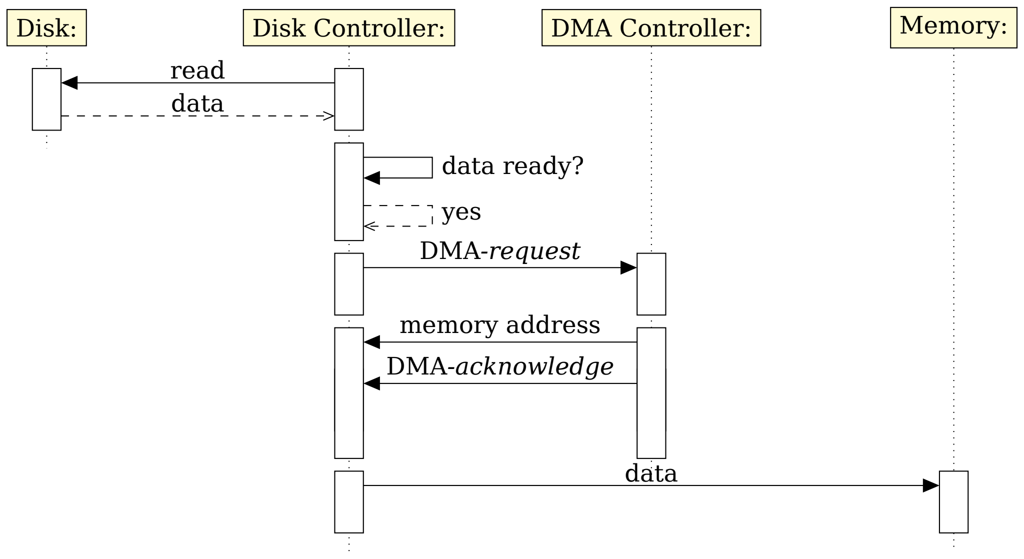


Fig. 35: DMA handshaking

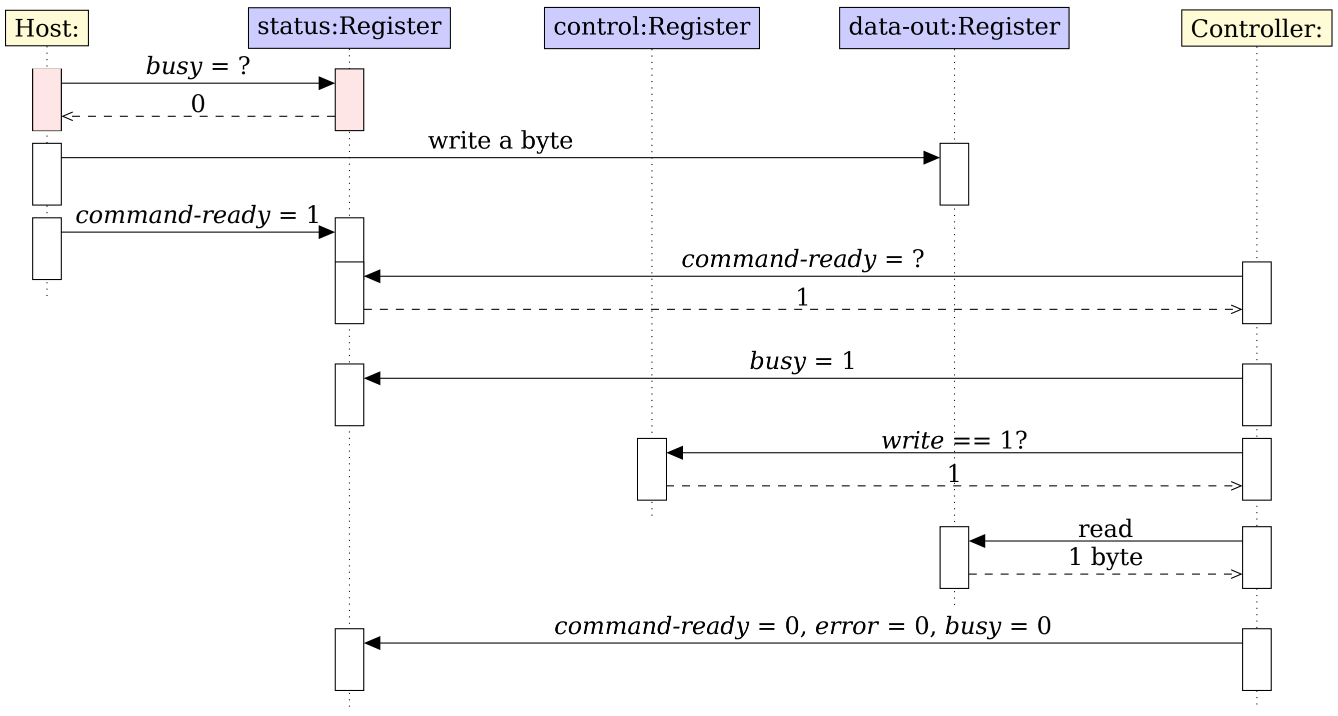


Fig. 36: Handshaking