List of Figures

1 Choosing an OS . . . 2

2 OS overview . . . 3

3 Motherboard chipsets . . . 4

4 Motherboard chipsets (bw version) . . . 4

5 CPU’s working cycle . . . 4

6 Bootstrapping . . . 5

7 Bootstrapping (bw version) . . . 5

8 Process’ virtual address space . . . 6

9 UNIX view of a process . . . 7

10 Process creation . . . 7

11 Thread operations . . . 7

12 Deadlock — Resource issues . . . 8

13 Deadlock notions . . . 8

14 Deadlock — Banker algorithm . . . 9

15 Deadlock avoidance . . . 9

16 Deadlock avoidance . . . 10

17 Producers and consumers . . . 10

18 A circular array . . . 10

19 Real mode memory layouts . . . 11

20 Tool chain . . . 12

21 Relocation . . . 13

22 First fit, best fit, worst fit . . . 14

23 Memory fragmentation . . . 14

24 Memory segmentation . . . 15

25 Memory segmentation — Address translation . . . 16

26 File system tables . . . 17

27 File tables . . . 17

28 File tables . . . 18

29 VFS objects . . . 18

30 VFS objects . . . 19

31 VFS file copy . . . 19

32 VFS file copy (bw version) . . . 19

33 DMA handshaking . . . 20

34 Handshaking . . . 20

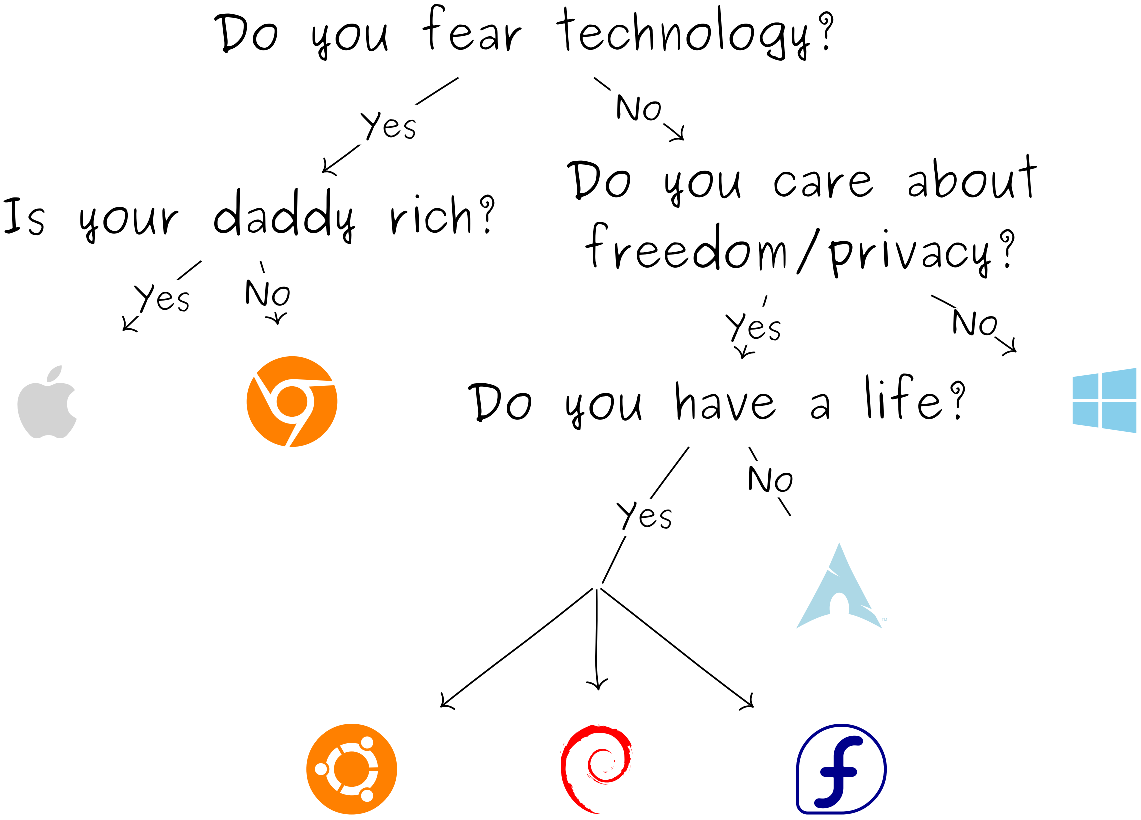


Fig. 1: Choosing an OS

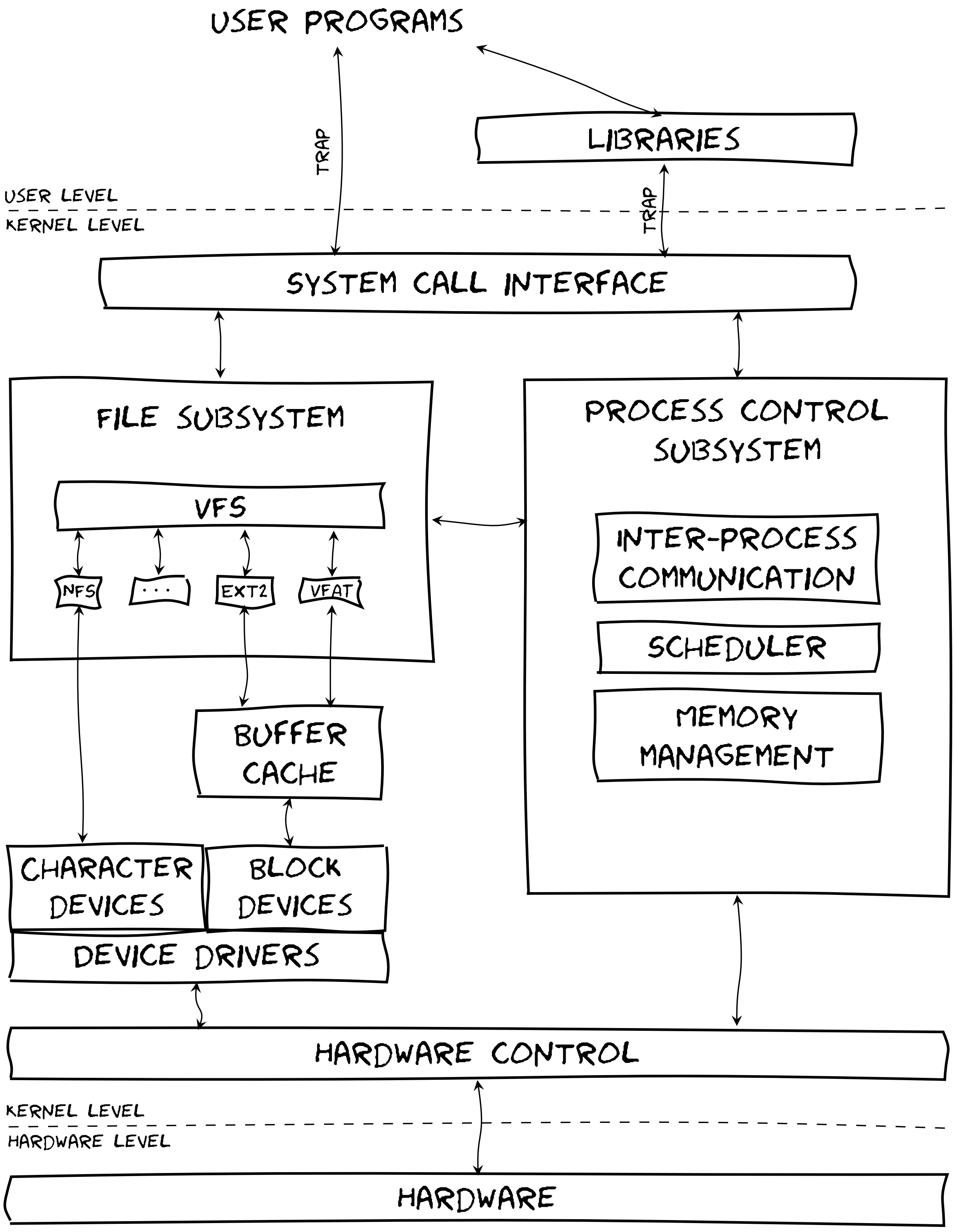


Fig. 2: OS overview

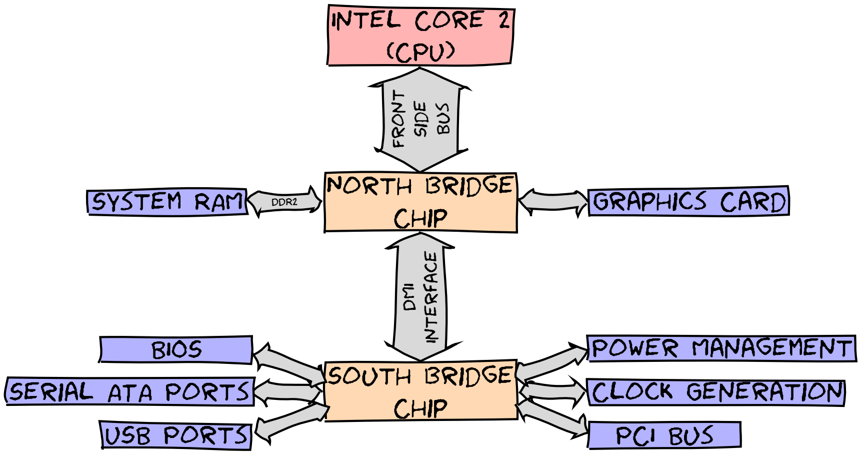


Fig. 3: Motherboard chipsets

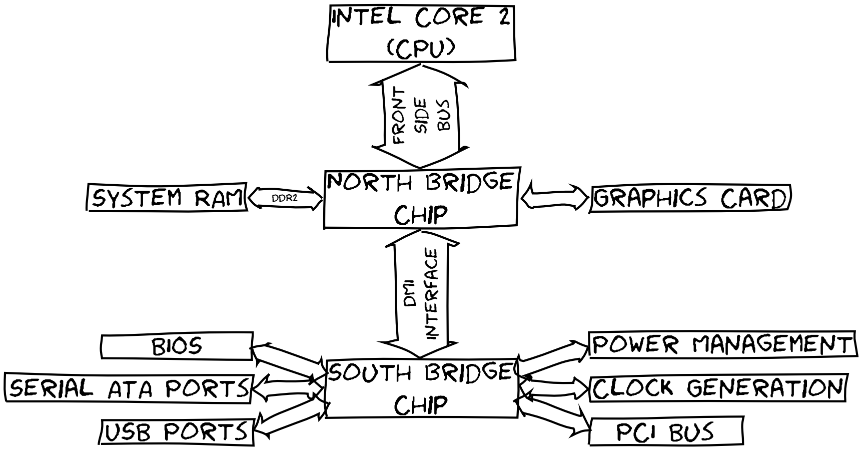


Fig. 4: Motherboard chipsets (bw version)

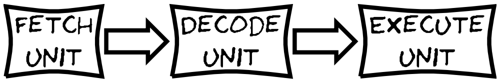


Fig. 5: CPU’s working cycle

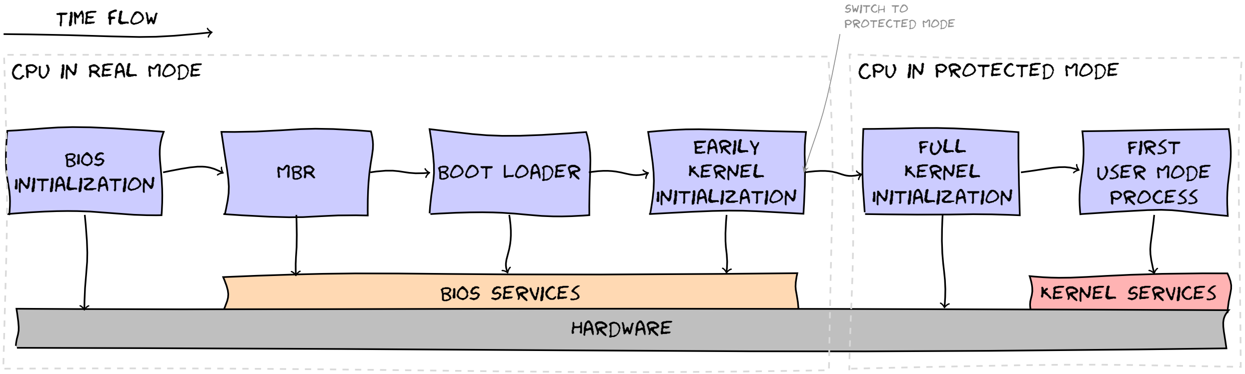


Fig. 6: Bootstrapping

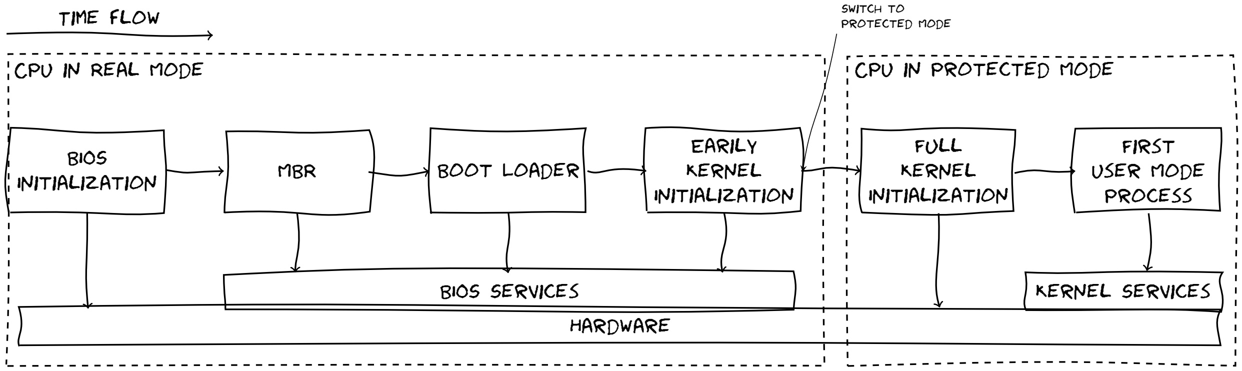


Fig. 7: Bootstrapping (bw version)

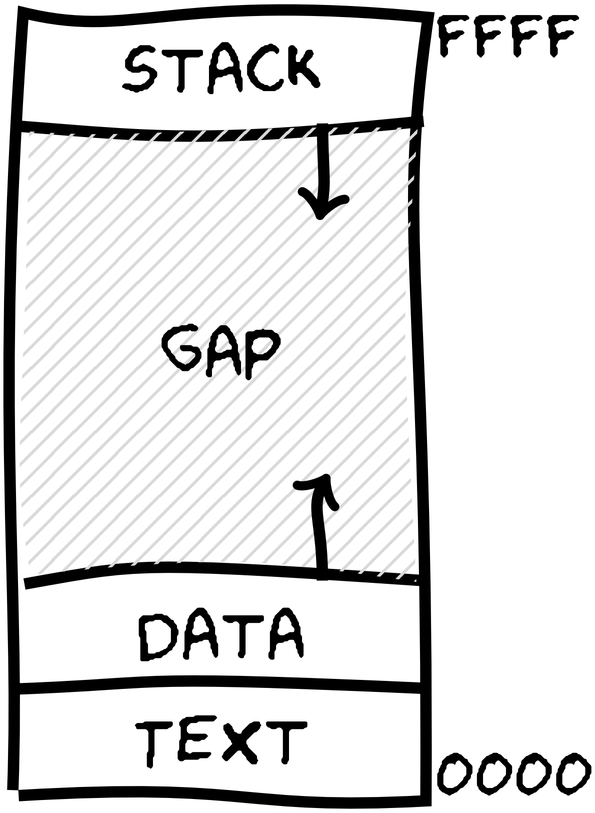
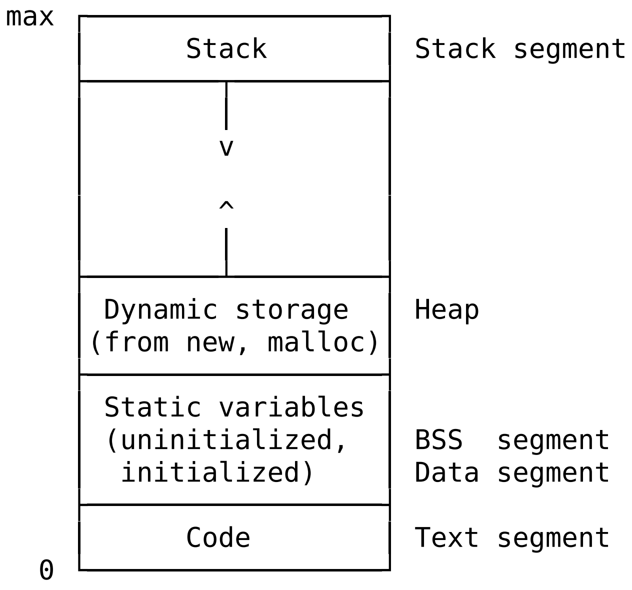


Fig. 8: Process' virtual address space



THE SIZE OF A PROCESS
(TEXT + DATA + BSS) IS
ESTABLISHED AT COMPILE TIME

Fig. 9: UNIX view of a process

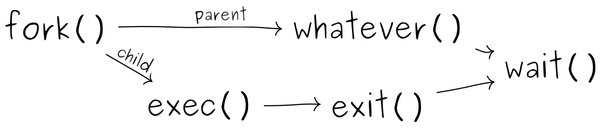


Fig. 10: Process creation

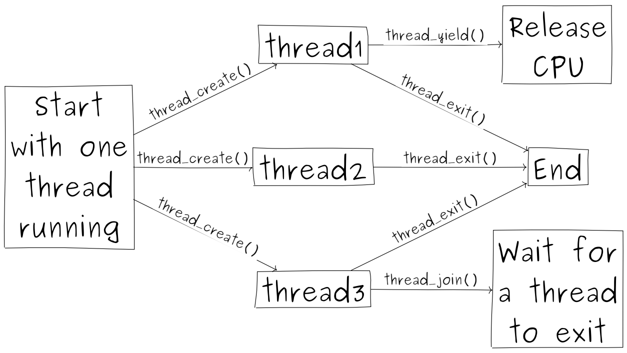


Fig. 11: Thread operations

```
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. 12: Deadlock — Resource issues

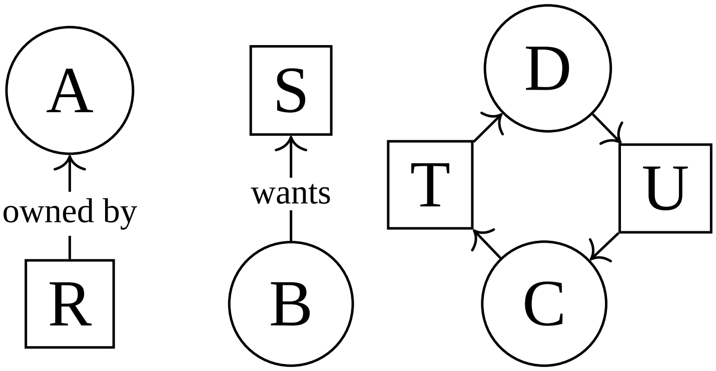


Fig. 13: Deadlock notions

	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. 14: Deadlock — Banker algorithm

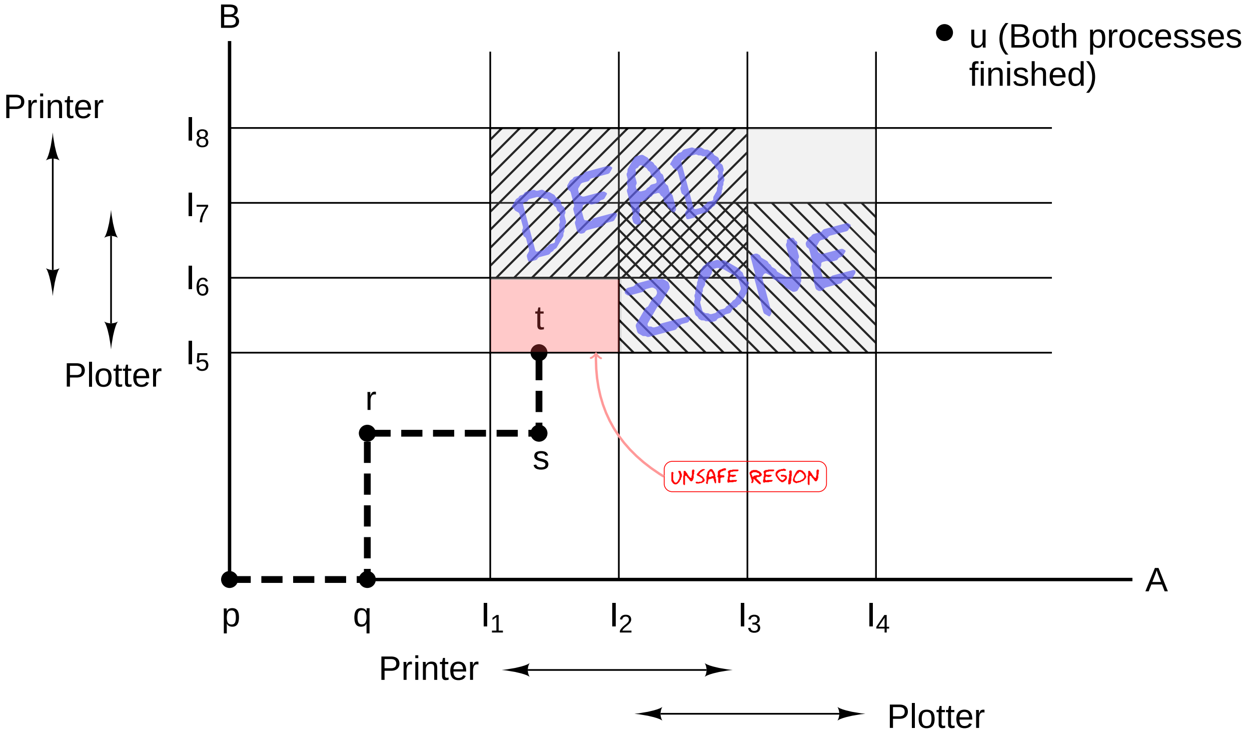
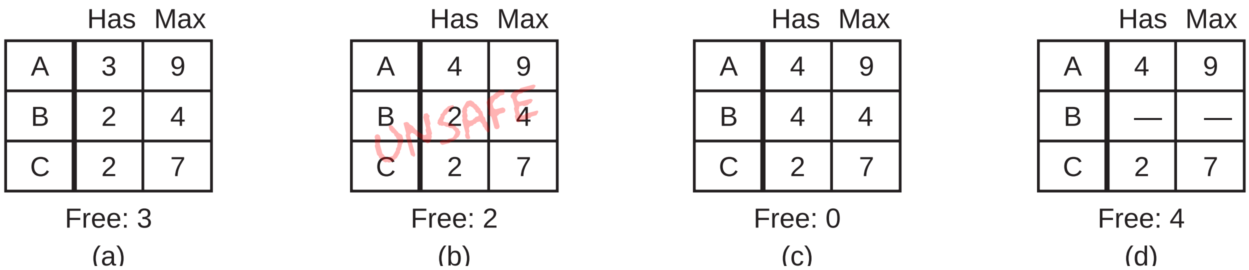


Fig. 15: Deadlock avoidance



	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

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. 16: Deadlock avoidance

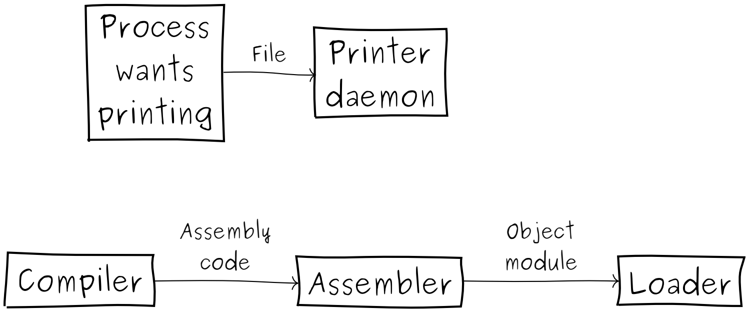


Fig. 17: Producers and consumers

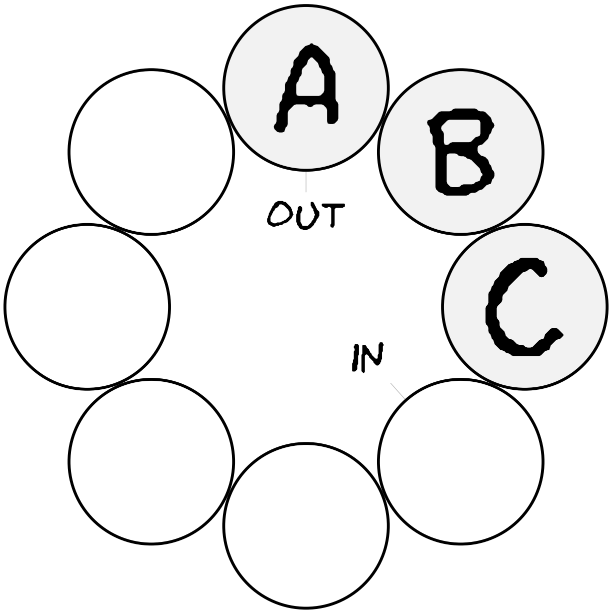


Fig. 18: A circular array

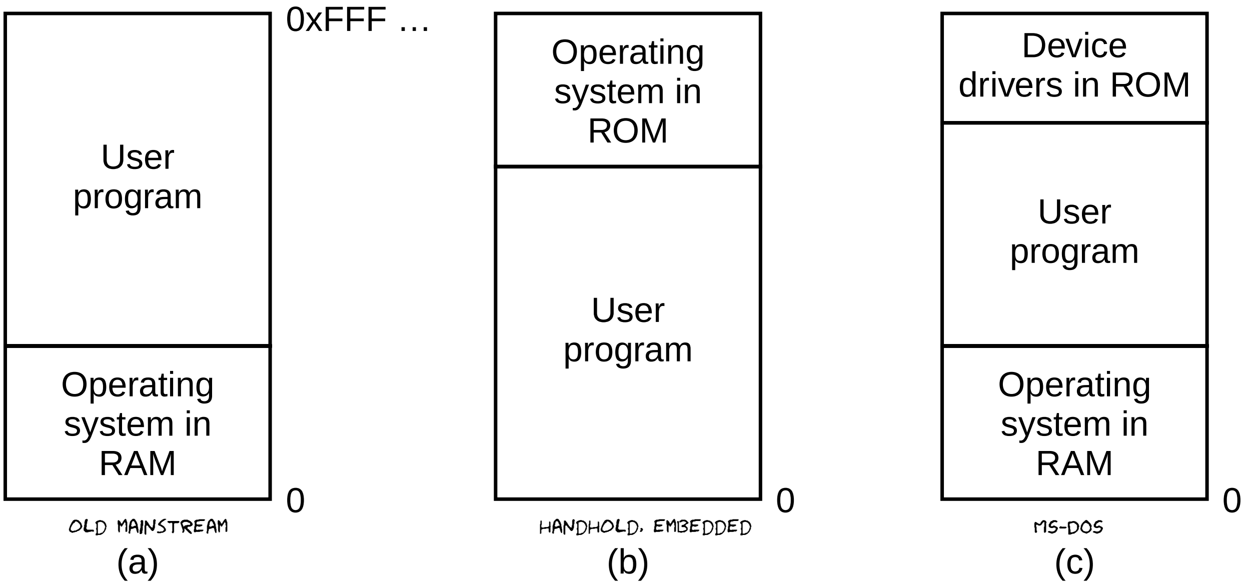
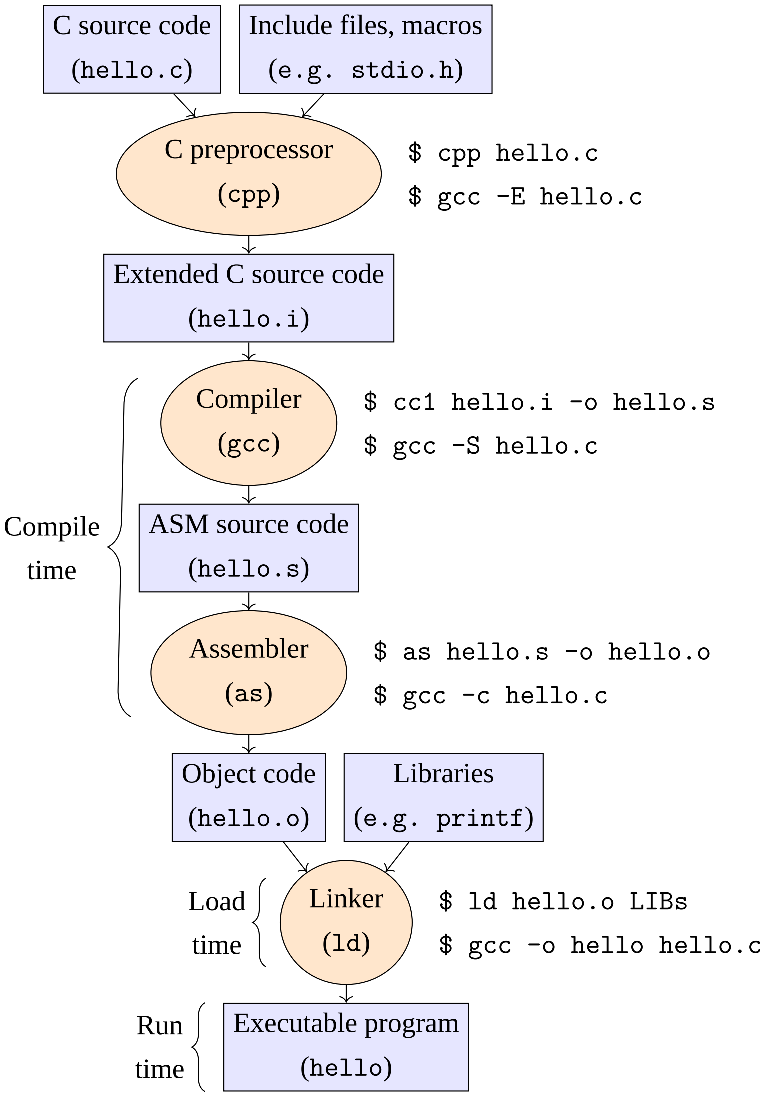


Fig. 19: Real mode memory layouts



```
$ cpp hello.c
$ gcc -E hello.c
```

```
$ cc1 hello.i -o hello.s
$ gcc -S hello.c
```

```
$ as hello.s -o hello.o
$ gcc -c hello.c
```

```
$ ld hello.o LIBs
$ gcc -o hello hello.c
```

Fig. 20: Tool chain

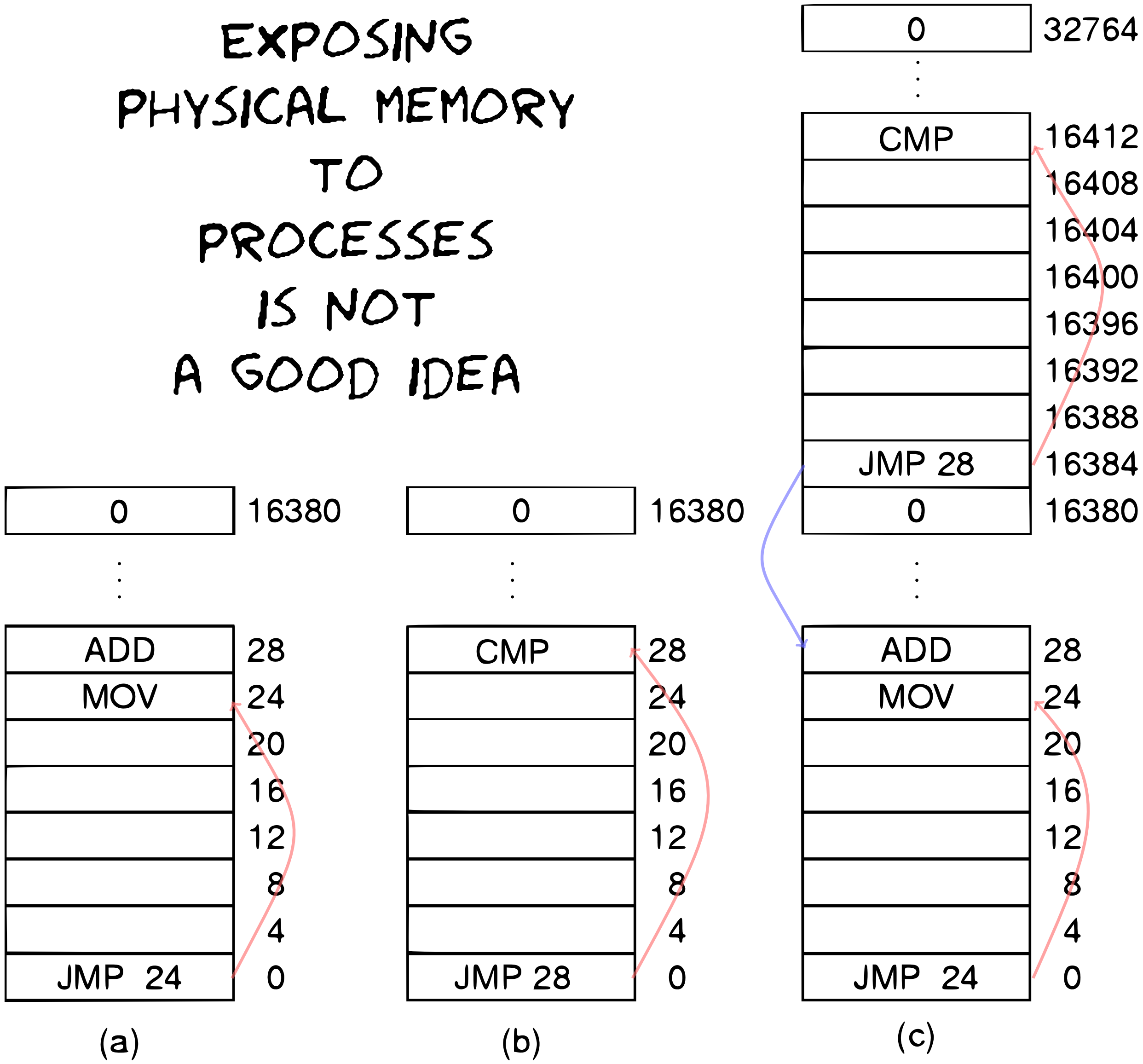


Fig. 21: Relocation

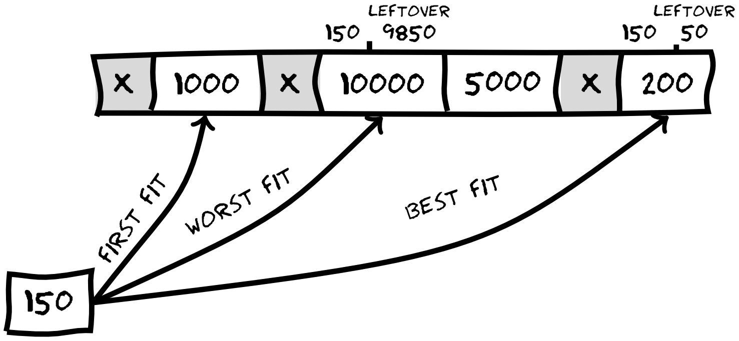


Fig. 22: First fit, best fit, worst fit

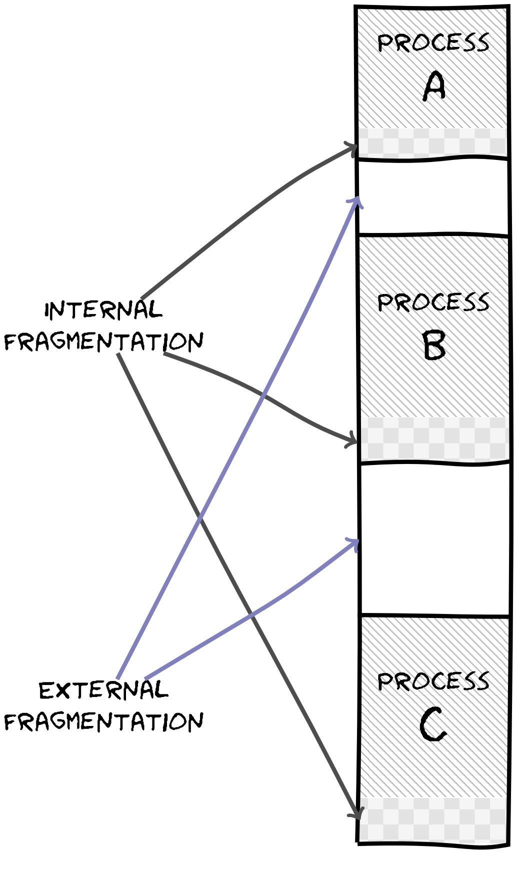


Fig. 23: Memory fragmentation

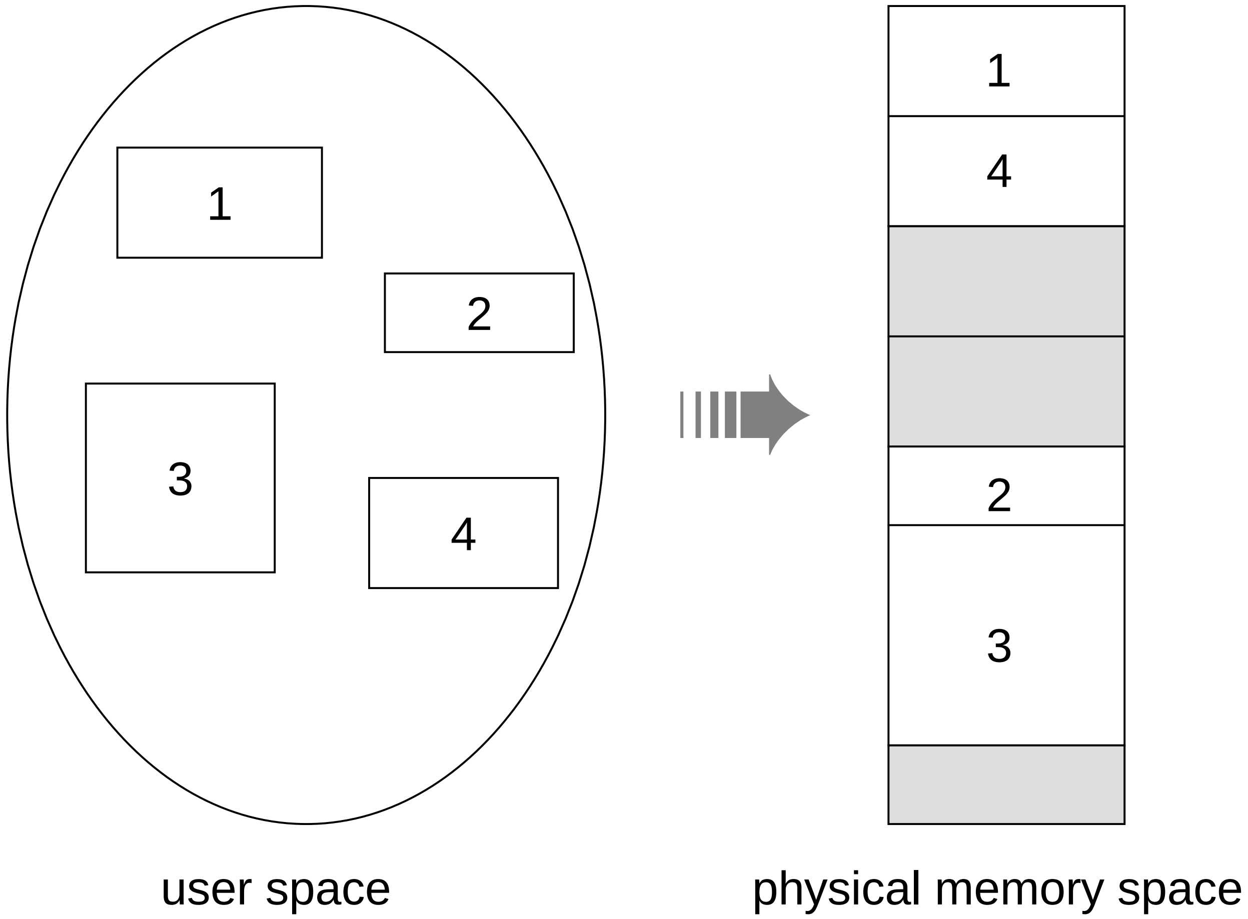
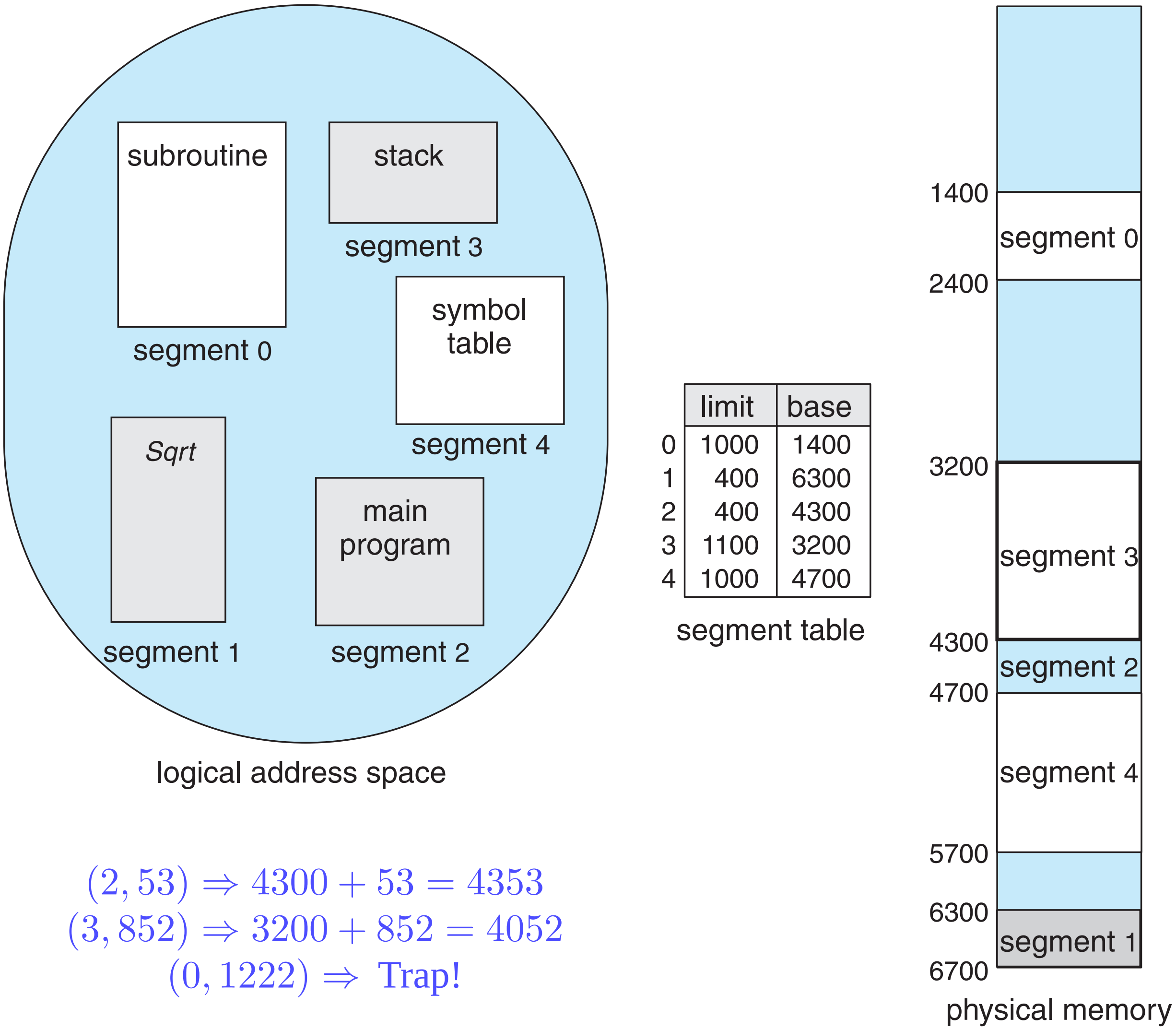


Fig. 24: Memory segmentation



	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. 25: Memory segmentation — Address translation

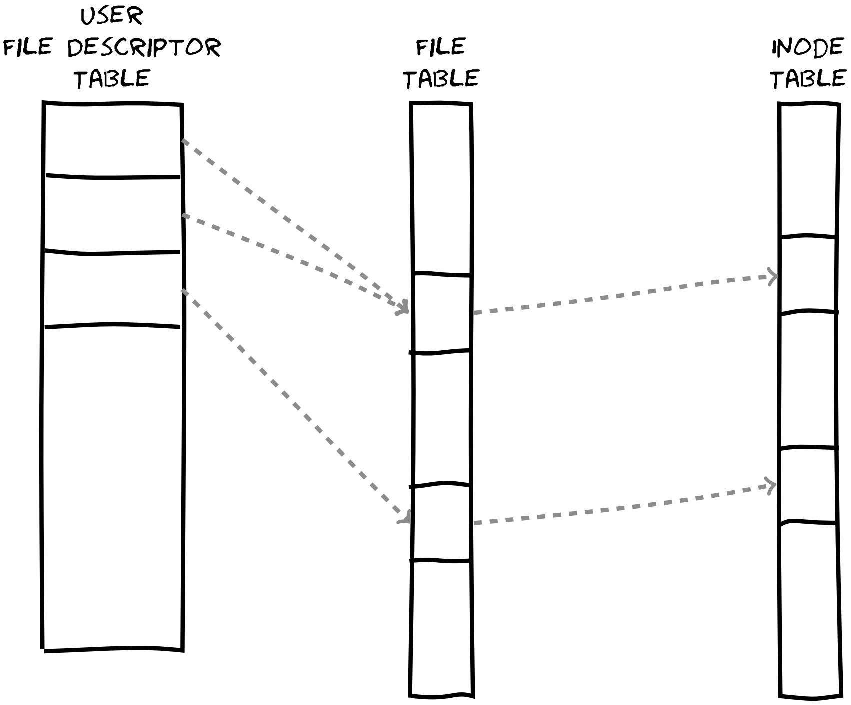


Fig. 26: File system tables

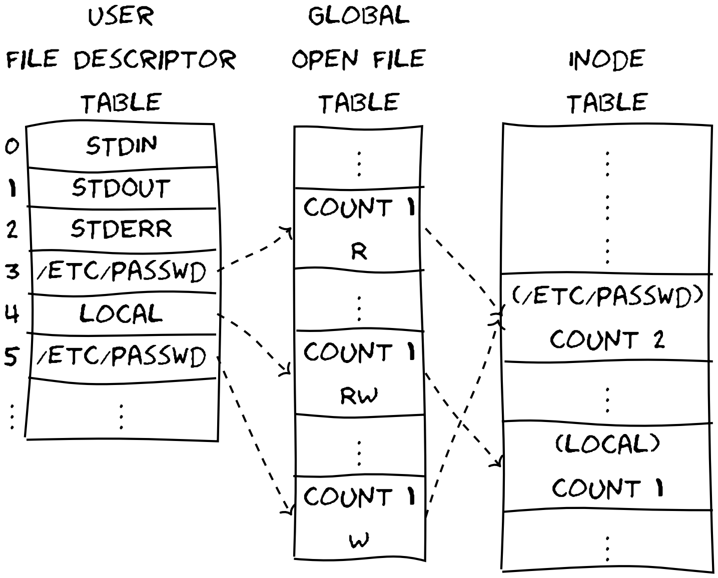


Fig. 27: File tables

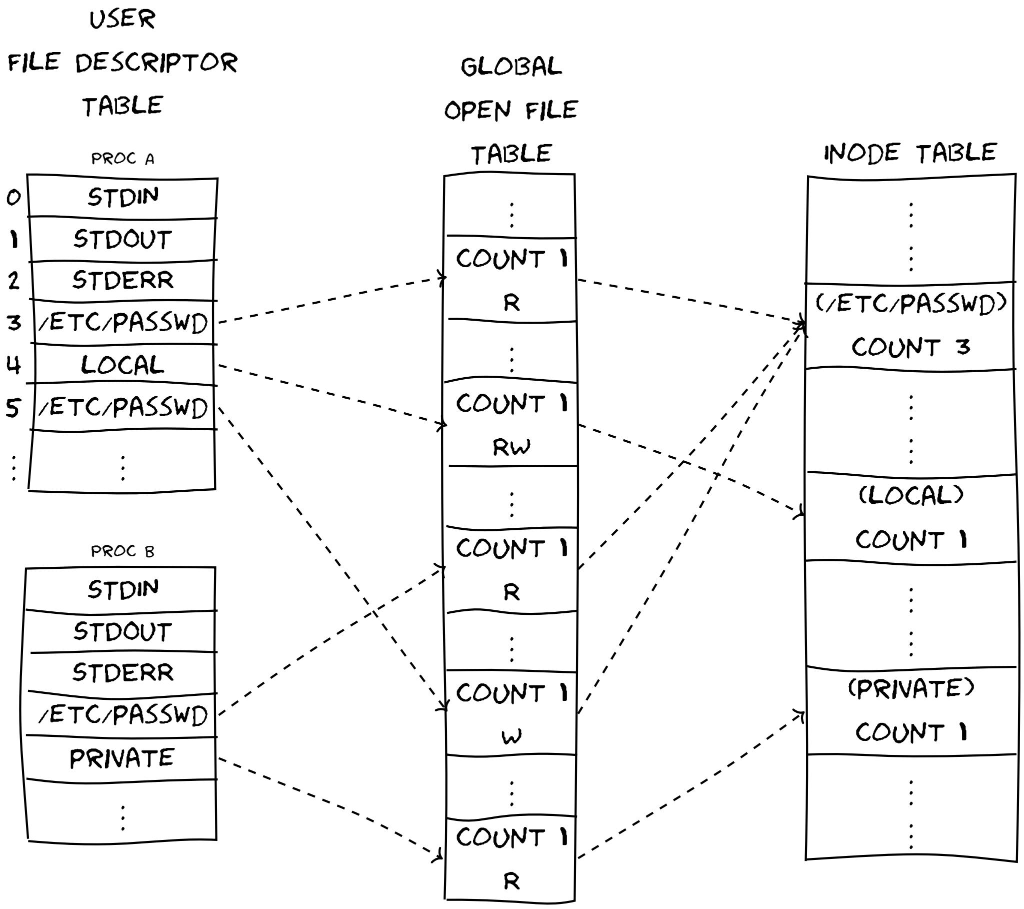


Fig. 28: File tables

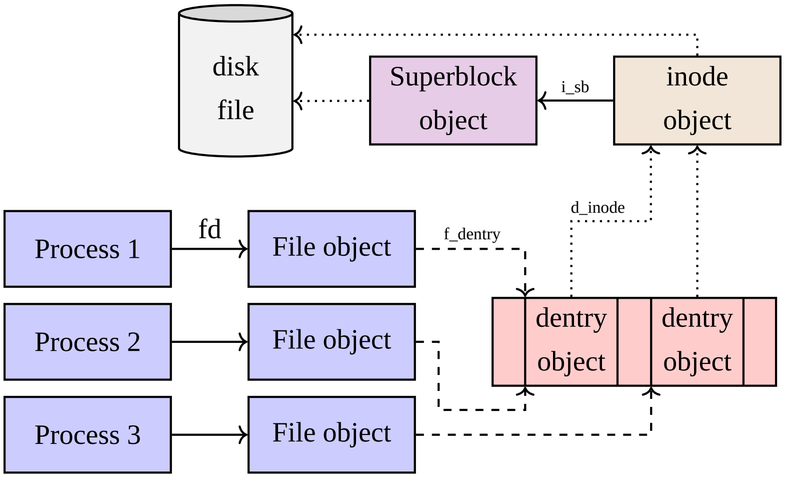


Fig. 29: VFS objects

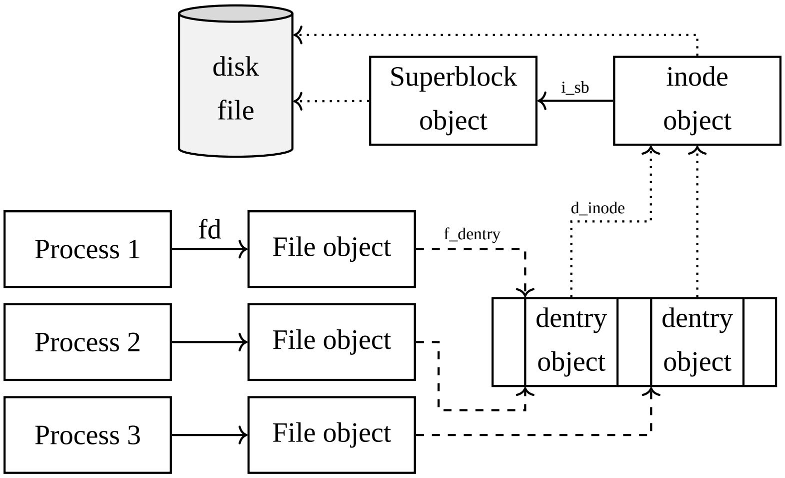


Fig. 30: VFS objects

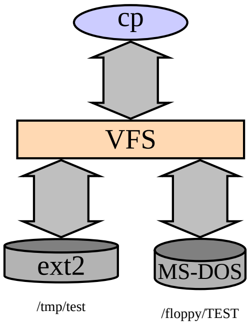


Fig. 31: VFS file copy

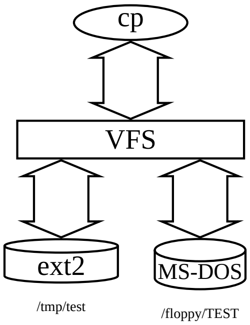


Fig. 32: VFS file copy (bw version)

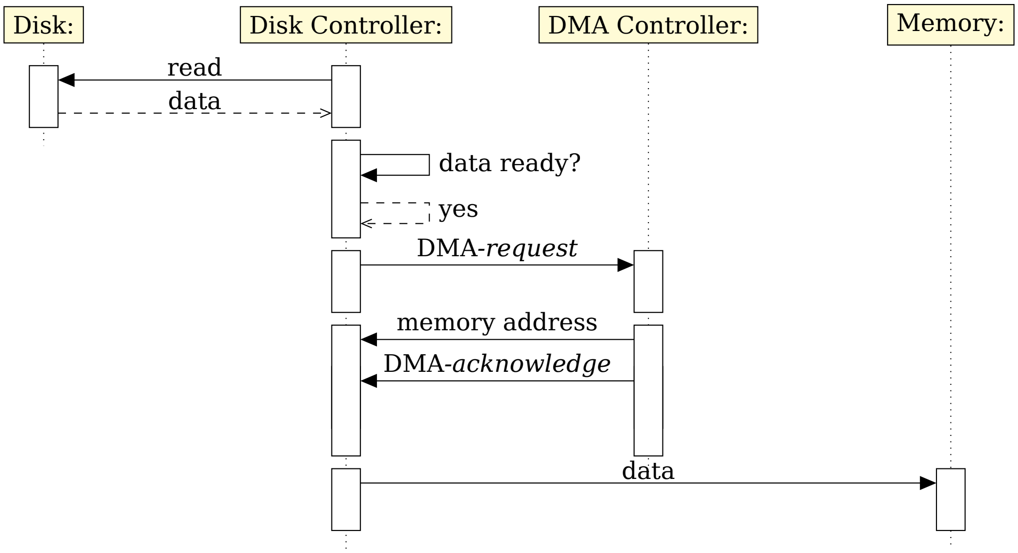


Fig. 33: DMA handshaking

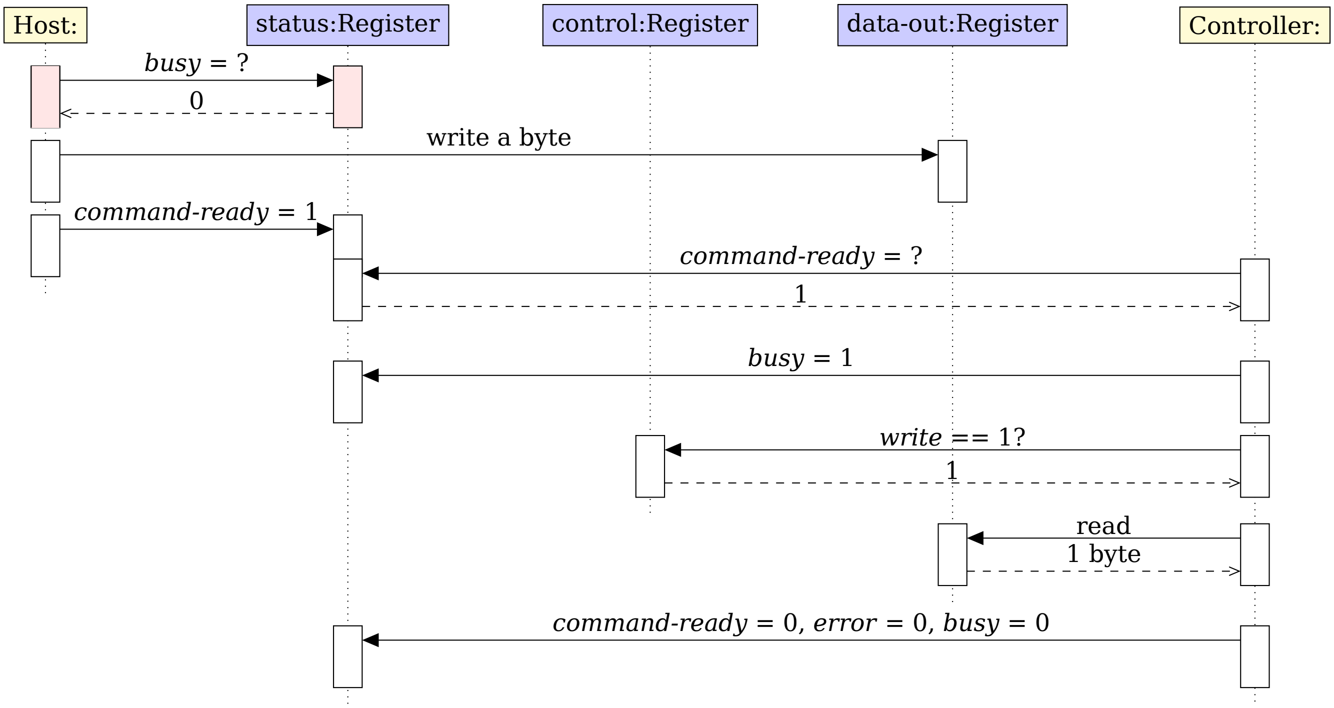


Fig. 34: Handshaking