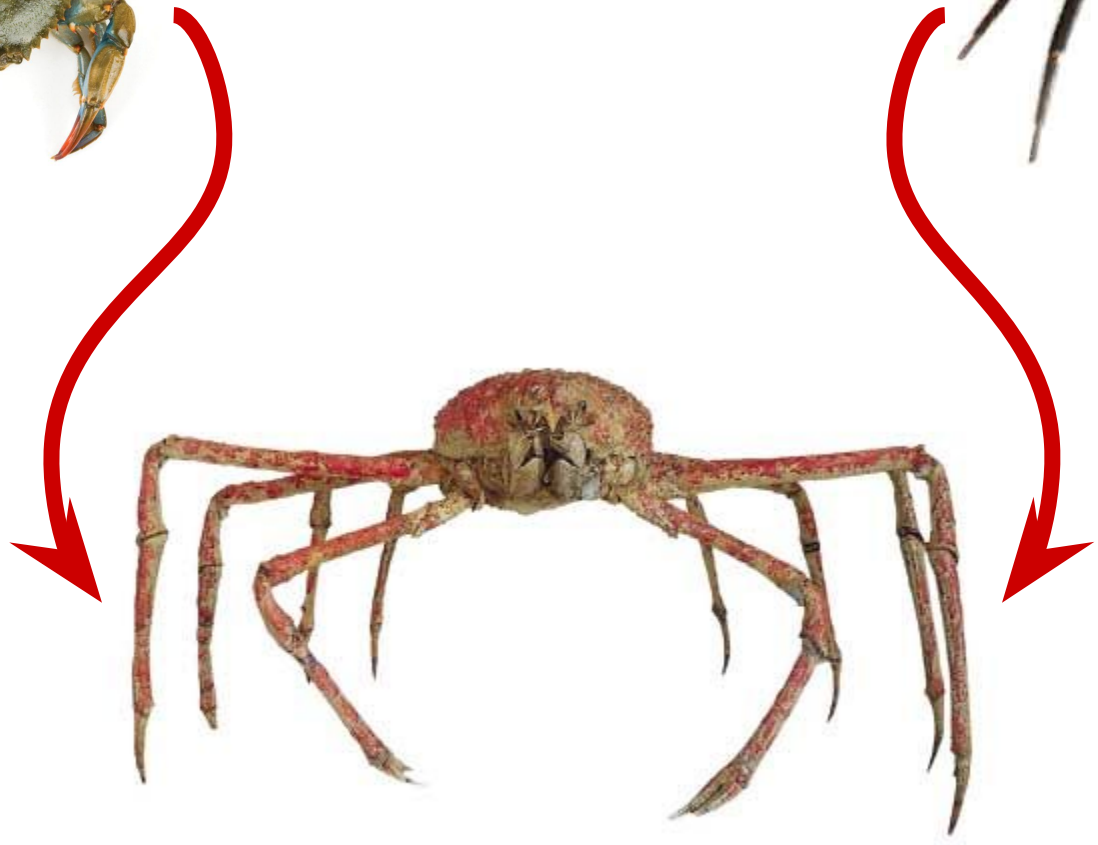
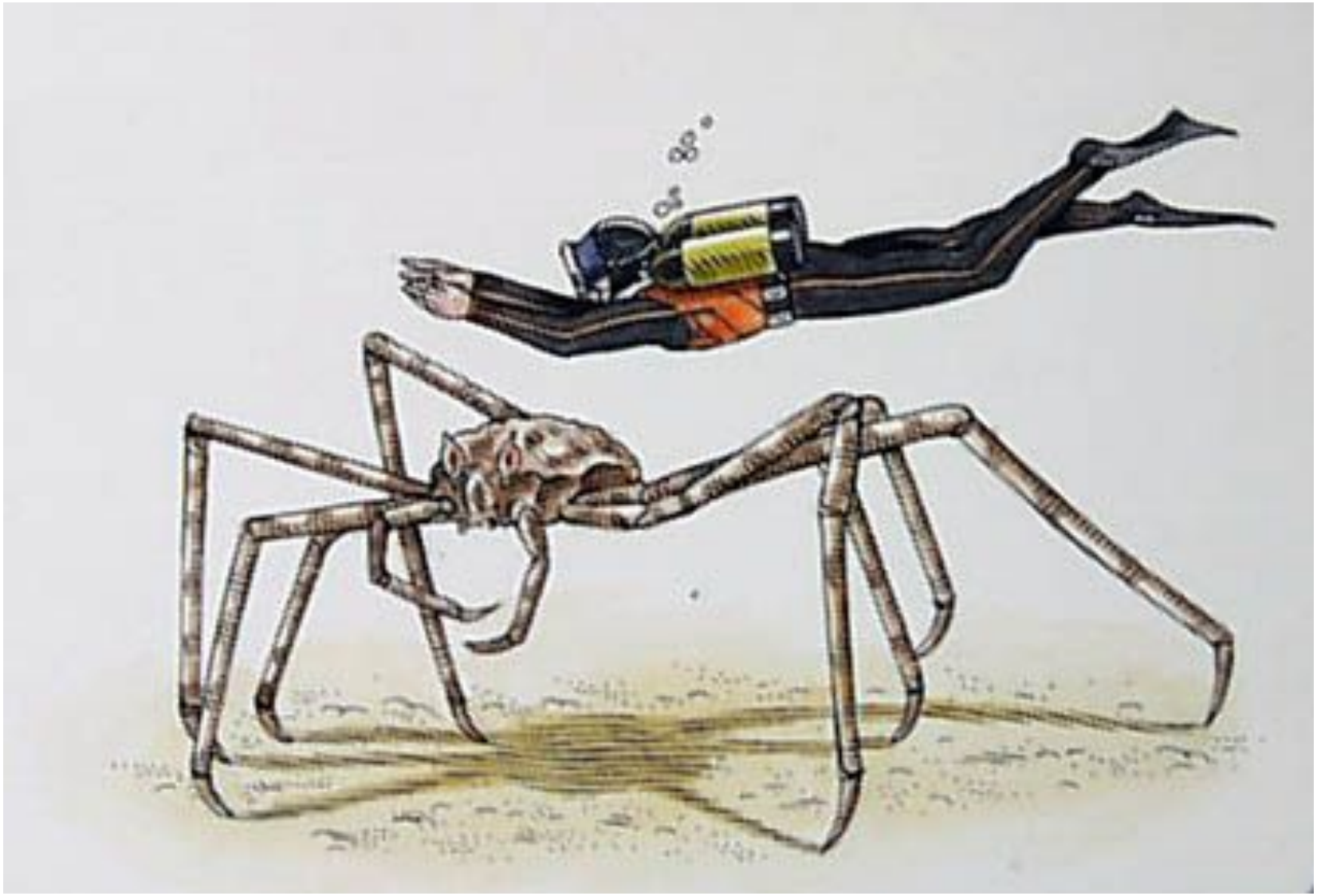
Crabzilla?

Japanese Spider Crab

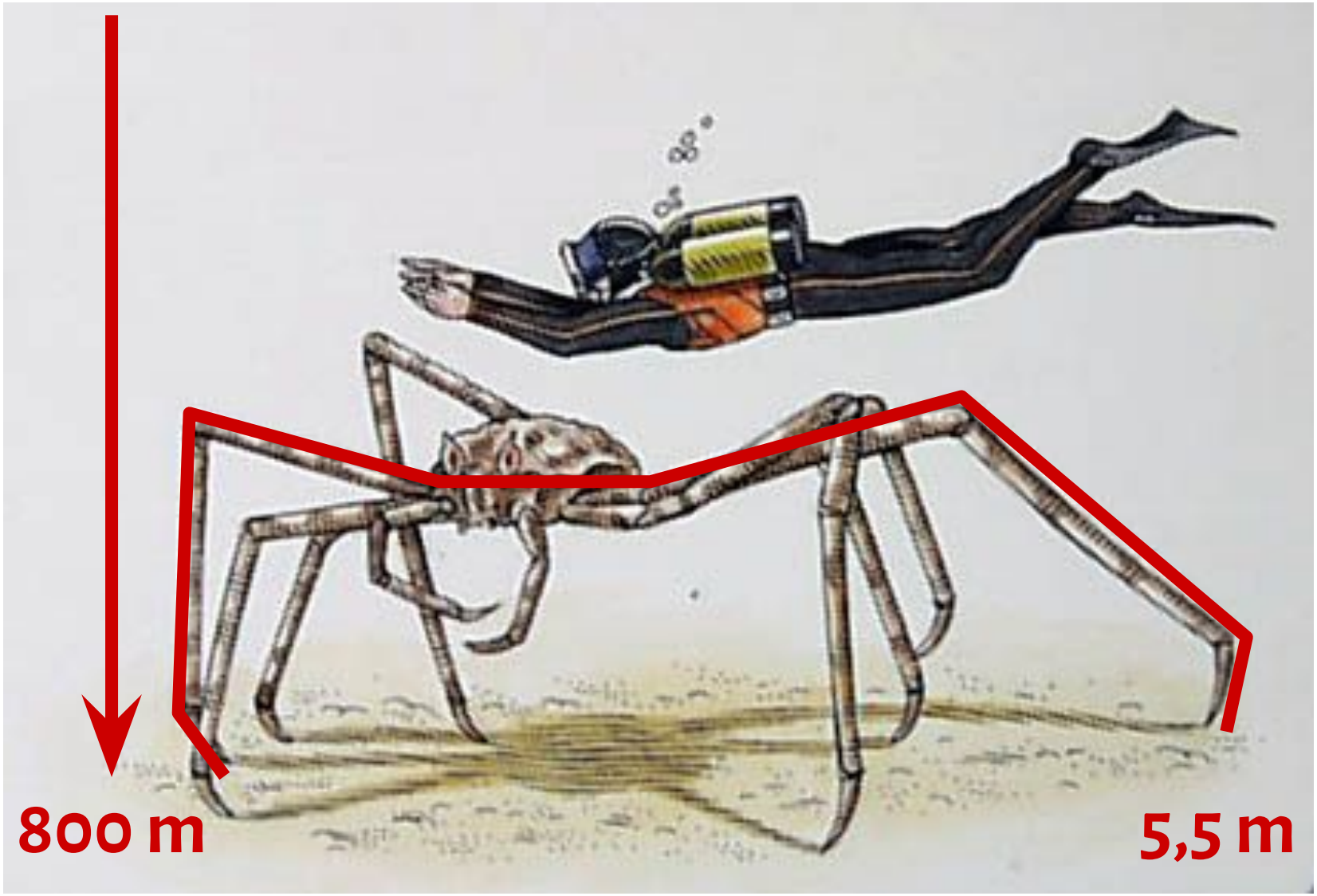
Japanese Spider Crab



Japanese Spider Crab



Japanese Spider Crab



Spider Crab Molting

Filesystems

Theory

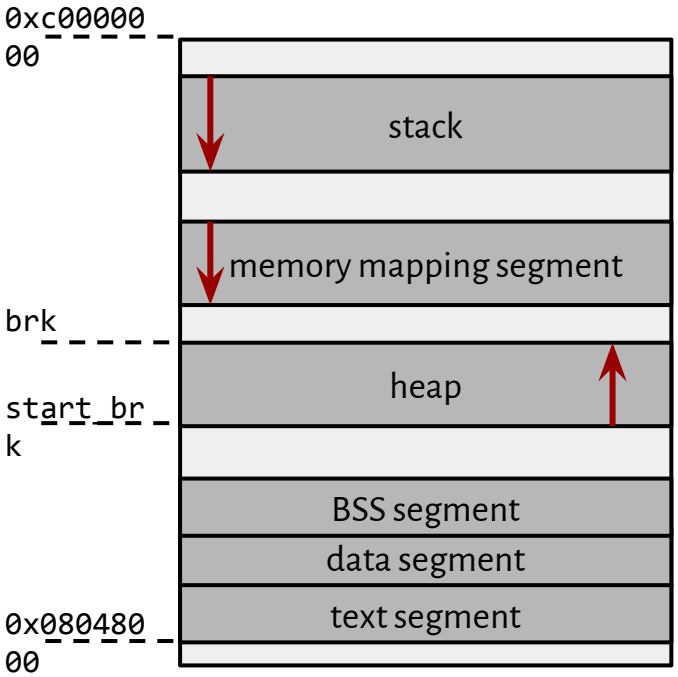
Practice

Assignments

A concept of a **process**

Remember?

a group of related resources



Andrew S. Tanenbaum: *Operating Systems Design and Implementation*, 3rd Edition; p. 64 - 68

File descriptors

File descriptors are userspace references to kernel objects.

Unix

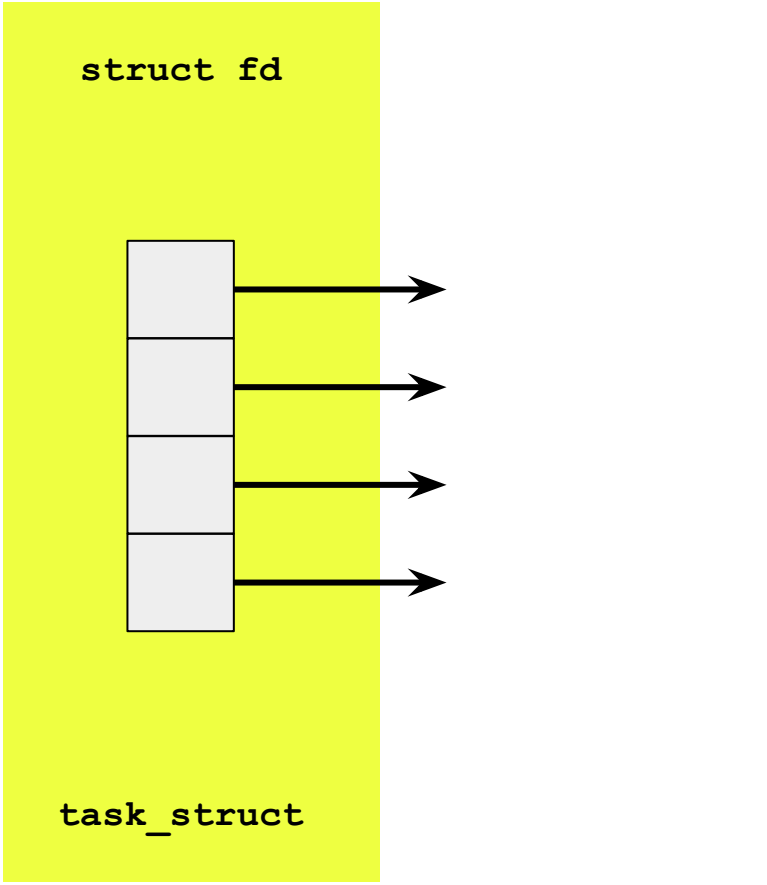
Everything is a file descriptor.

File descriptors

File descriptors are userspace references to kernel objects.

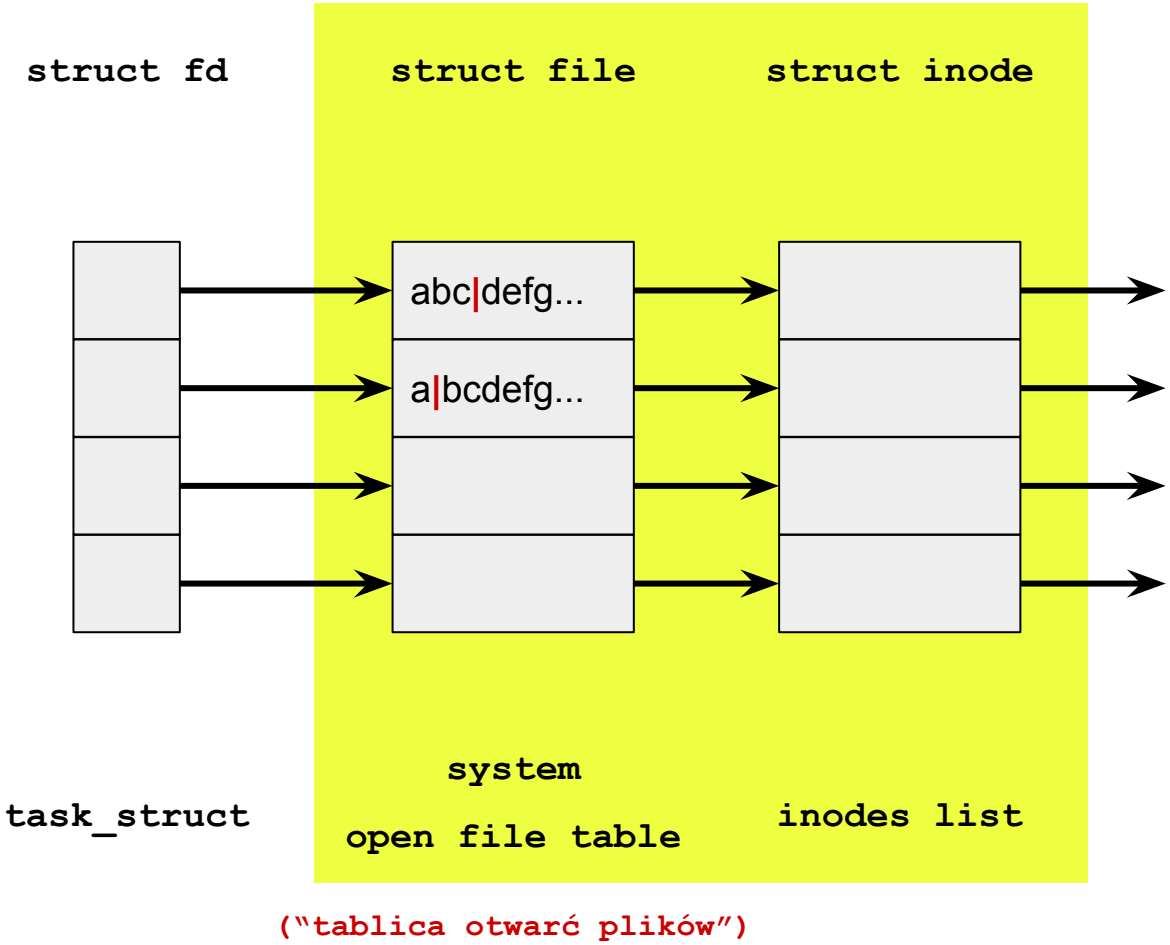
File descriptors

File descriptors are userspace references to kernel objects.



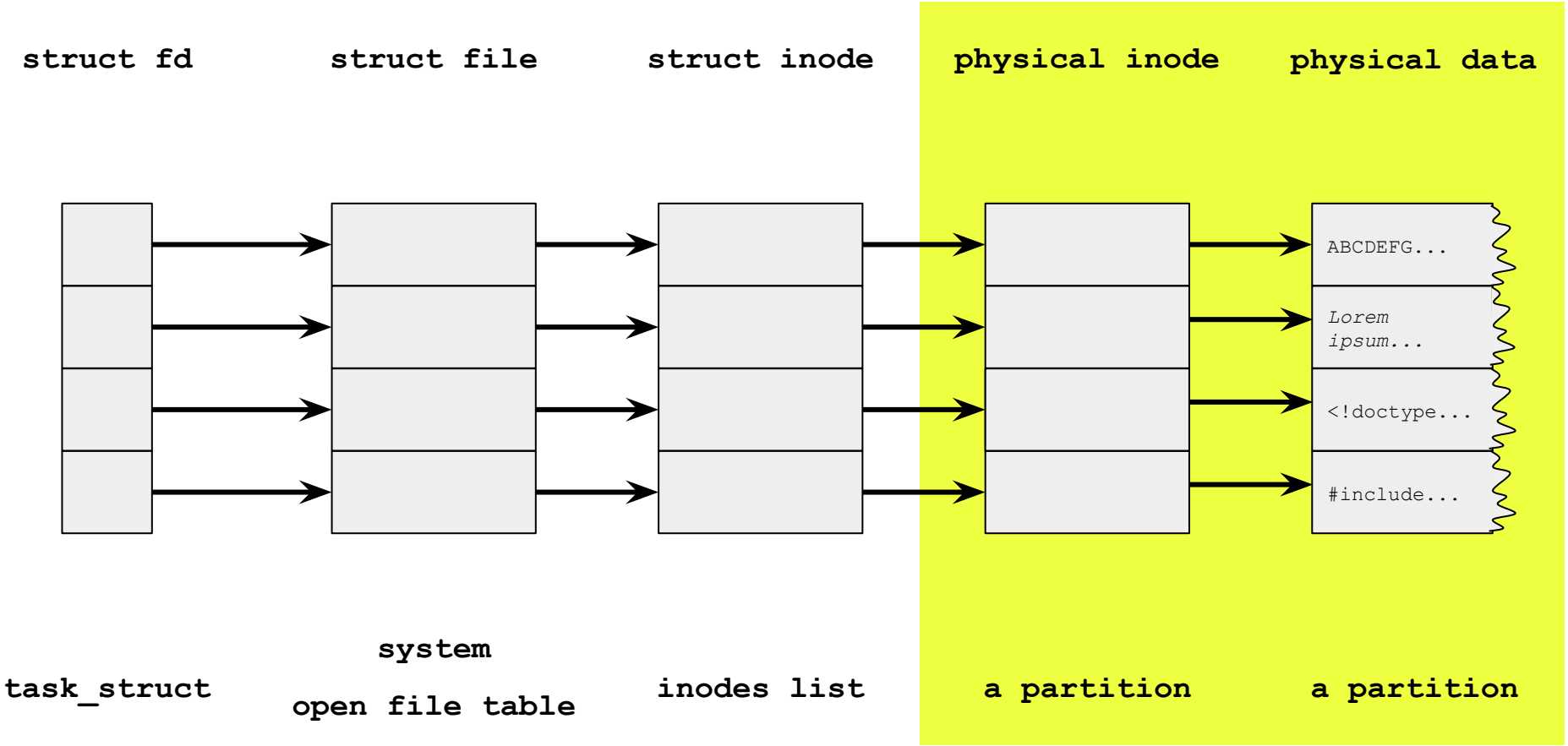
File descriptors

File descriptors are userspace references to kernel objects.



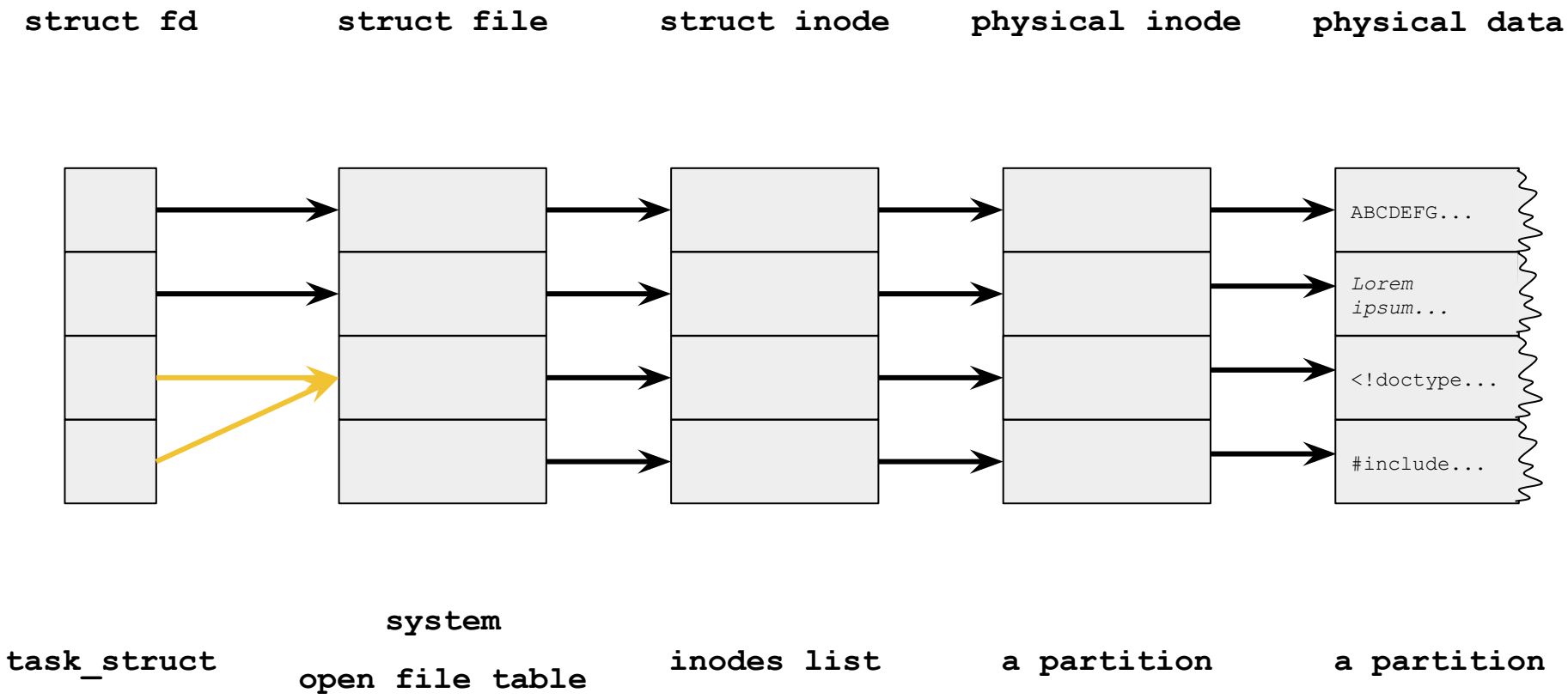
File descriptors

File descriptors are userspace references to kernel objects.



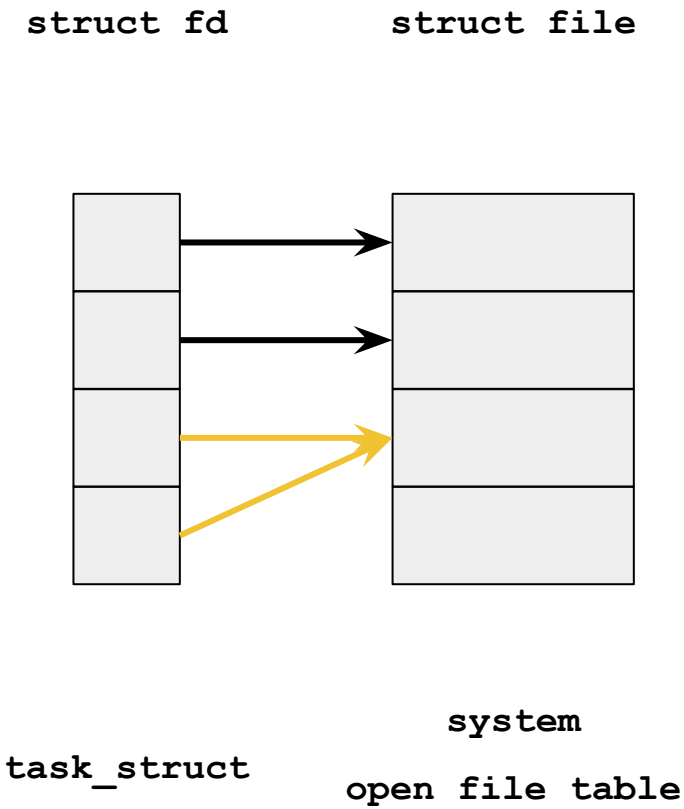
File descriptors

File descriptors are userspace references to kernel objects.



File descriptors

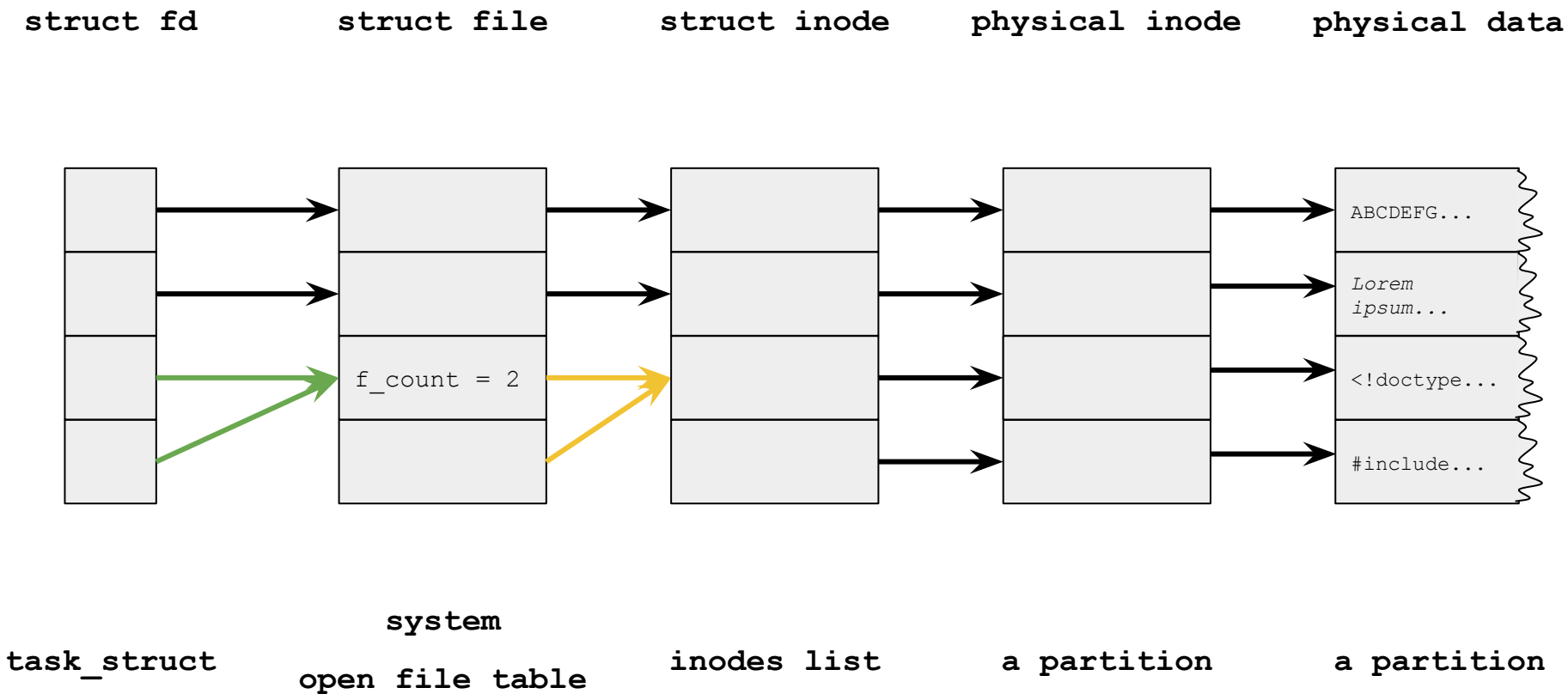
File descriptors are userspace references to kernel objects.



- ★ syscall `dup` - returns a new fd that refers to the same connection
- ★ syscall fork - children "inherit" descriptors from their parent process

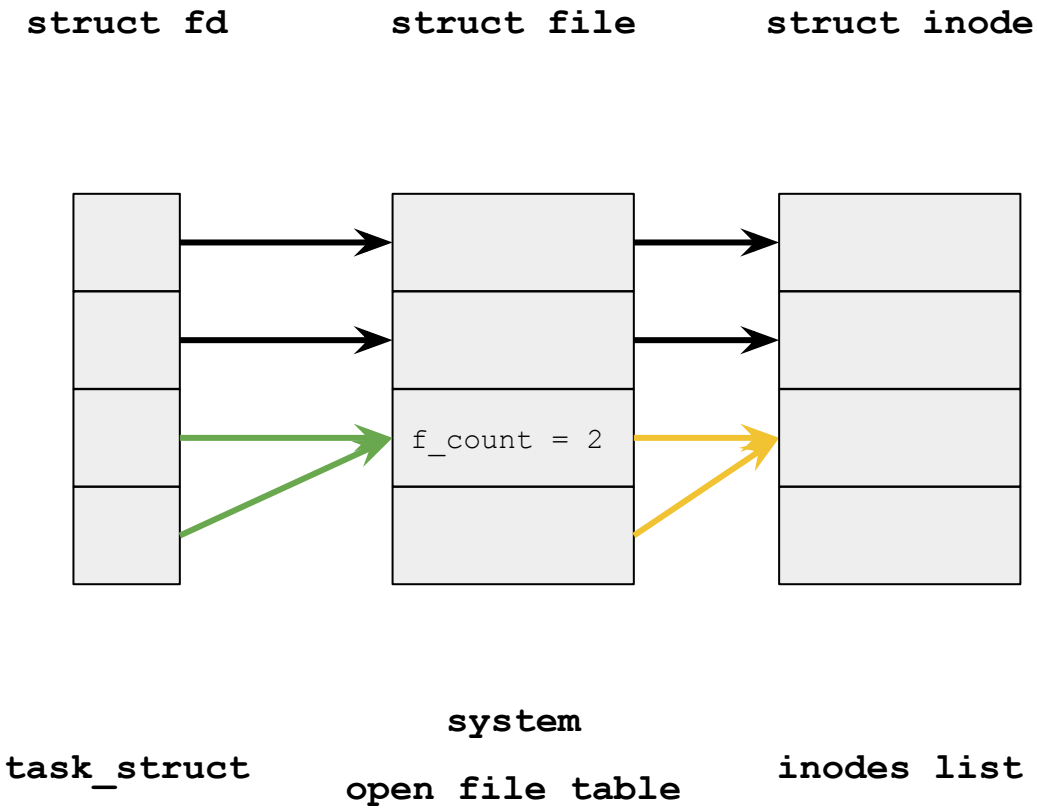
File descriptors

File descriptors are userspace references to kernel objects.



File descriptors

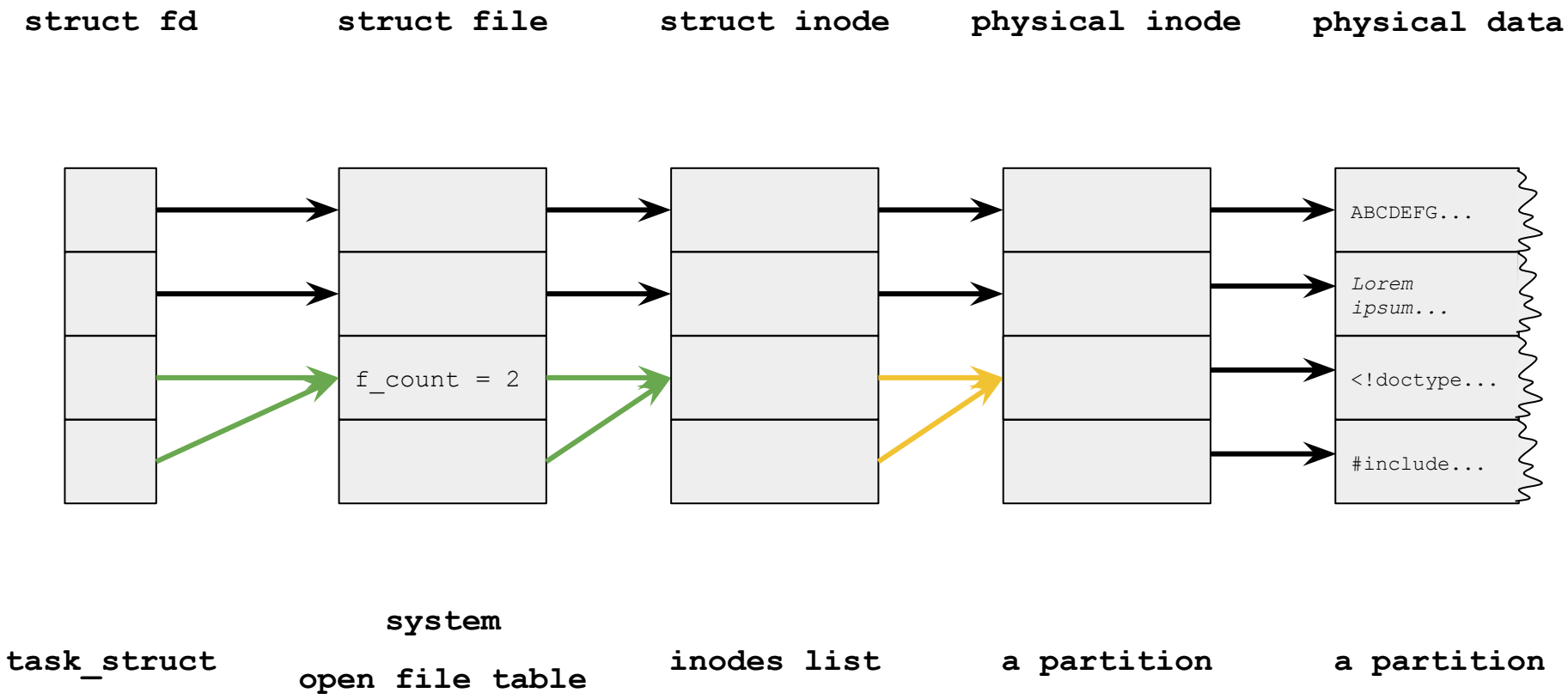
File descriptors are userspace references to kernel objects.



★ a single file can be accessed many times in parallel

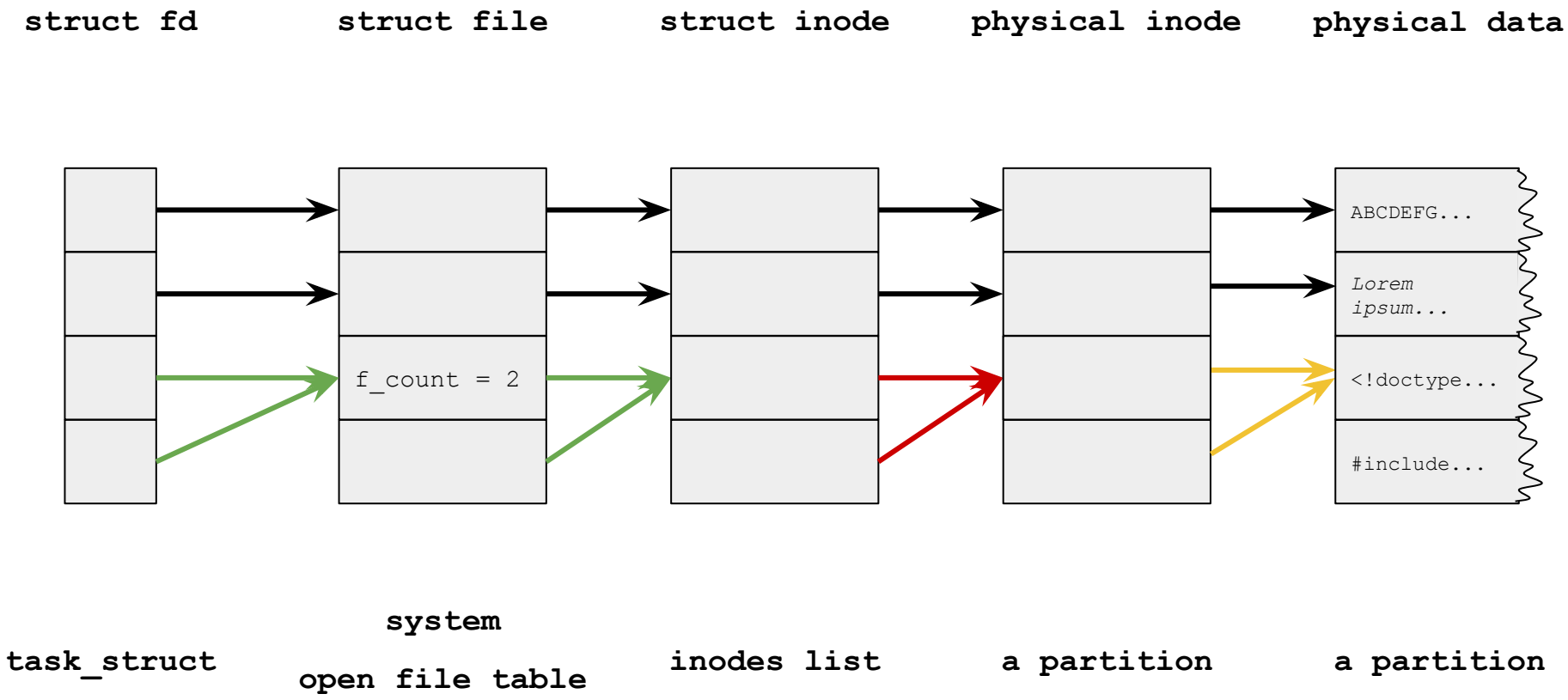
File descriptors

File descriptors are userspace references to kernel objects.



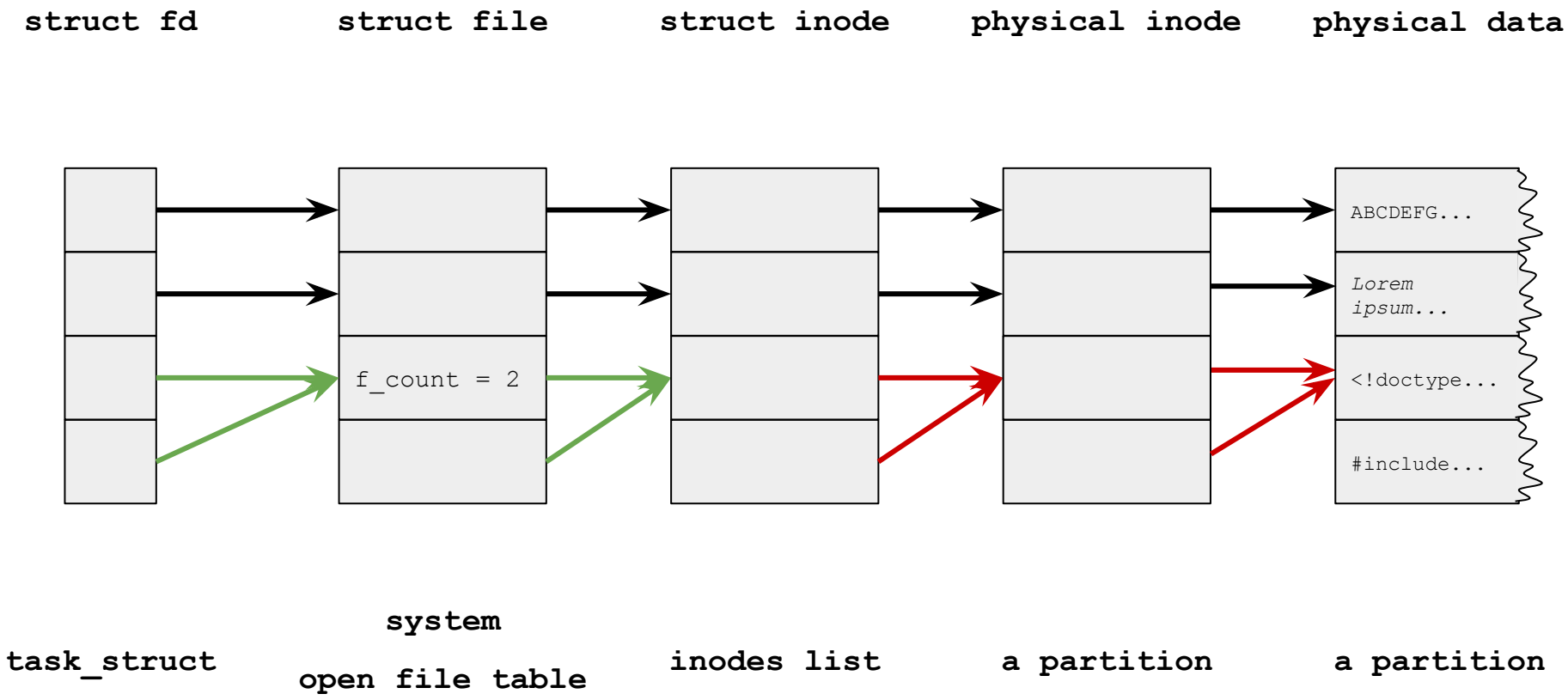
File descriptors

File descriptors are userspace references to kernel objects.



File descriptors

File descriptors are userspace references to kernel objects.



File descriptors

```
$ cat /home/inga/test.txt
abcdefg
```

```
x = open("/home/inga/test.txt", O_RDONLY);
```

File descriptors

```
$ cat /home/inga/test.txt
abcdefg
```

```
x = open("/home/inga/test.txt", O_RDONLY);
x2 = dup(x);
```

File descriptors

```
$ cat /home/inga/test.txt
abcdefg
```

```
x = open("/home/inga/test.txt", O_RDONLY);
x2 = dup(x);
y = open("/home/inga/test.txt", O_RDONLY);
```

File descriptors

```
$ cat /home/inga/test.txt
abcdefg
```

```
x = open("/home/inga/test.txt", O_RDONLY);
x2 = dup(x);
y = open("/home/inga/test.txt", O_RDONLY);
read(x, &c, 1);
putchar(c);
```

File descriptors

```
$ cat /home/inga/test.txt
abcdefg
```

```
x = open("/home/inga/test.txt", O_RDONLY);
x2 = dup(x);
y = open("/home/inga/test.txt", O_RDONLY);
read(x, &c, 1);          // "a"
putchar(c);
```

File descriptors

```
$ cat /home/inga/test.txt
abcdefg
```

```
x = open("/home/inga/test.txt", O_RDONLY);
x2 = dup(x);
y = open("/home/inga/test.txt", O_RDONLY);
read(x, &c, 1);          // "a"
putchar(c);
read(x2, &c, 1);
putchar(c);
```

File descriptors

```
$ cat /home/inga/test.txt
abcdefg
```

```
x = open("/home/inga/test.txt", O_RDONLY);
x2 = dup(x);
y = open("/home/inga/test.txt", O_RDONLY);
read(x, &c, 1);          // "a"
putchar(c);
read(x2, &c, 1);         // "b"
putchar(c);
```

File descriptors

```
$ cat /home/inga/test.txt
abcdefg
```

```
x = open("/home/inga/test.txt", O_RDONLY);
x2 = dup(x);
y = open("/home/inga/test.txt", O_RDONLY);
read(x, &c, 1);          // "a"
putchar(c);
read(x2, &c, 1);         // "b"
putchar(c);
read(y, &c, 1);
putchar(c);
```

File descriptors

```
$ cat /home/inga/test.txt
abcdefg
```

```
x = open("/home/inga/test.txt", O_RDONLY);
x2 = dup(x);
y = open("/home/inga/test.txt", O_RDONLY);
read(x, &c, 1);        // "a"
putchar(c);
read(x2, &c, 1);       // "b"
putchar(c);
read(y, &c, 1);        // "a"
putchar(c);
```

Zadanie A1

Załóżmy, że pewien proces, który **nie tworzył procesów potomnych**, korzysta z ***n*** otwartych plików.

Zadanie A1

Załóżmy, że pewien proces, który **nie tworzył procesów potomnych**, korzysta z ***n*** otwartych plików.

Wszystkie te **pliki są różne** (odpowiadają im różne i-węzły).

Zadanie A1

Załóżmy, że pewien proces, który **nie tworzył procesów potomnych**, korzysta z ***n*** otwartych plików.

Wszystkie te **pliki są różne** (odpowiadają im różne i-węzły).

Załóżmy również, że ten proces jest **jedynym procesem w systemie, który ma otwarte pliki**.

Zadanie A1

Załóżmy, że pewien proces, który **nie tworzył procesów potomnych**, korzysta z ***n*** otwartych plików.
Wszystkie te **pliki są różne** (odpowiadają im różne i-węzły).
Załóżmy również, że ten proces jest **jedynym procesem w systemie, który ma otwarte pliki**.

Czy jest możliwa sytuacja, w której liczba deskryptorów otwartych plików dla tego procesu jest:

- mniejsza ostro od ***n***?
- większa ostro od ***n***?

Zadanie A1

Załóżmy, że pewien proces, który **nie tworzył procesów potomnych**, korzysta z ***n*** otwartych plików.
Wszystkie te **pliki są różne** (odpowiadają im różne i-węzły).
Załóżmy również, że ten proces jest **jedynym procesem w systemie, który ma otwarte pliki**.

Czy jest możliwa sytuacja, w której liczba deskryptorów otwartych plików dla tego procesu jest:

- mniejsza ostro od ***n***?
- większa ostro od ***n***?

Nie może być mniejsza od ***n***.

Zadanie A1

Załóżmy, że pewien proces, który **nie tworzył procesów potomnych**, korzysta z ***n*** otwartych plików.
Wszystkie te **pliki są różne** (odpowiadają im różne i-węzły).
Załóżmy również, że ten proces jest **jedynym procesem w systemie, który ma otwarte pliki**.

Czy jest możliwa sytuacja, w której liczba deskryptorów otwartych plików dla tego procesu jest:

- mniejsza ostro od ***n***?
- większa ostro od ***n***?

Może być większa od ***n***, jeśli proces otworzy ten sam plik wielokrotnie albo użyje syscalla duplicate: dup().

Zadanie A1

Załóżmy, że pewien proces, który **nie tworzył procesów potomnych**, korzysta z ***n*** otwartych plików.
Wszystkie te **pliki są różne** (odpowiadają im różne i-węzły).
Załóżmy również, że ten proces jest **jedynym procesem w systemie, który ma otwarte pliki**.

Opisz, jak można taką sytuację wykryć, znając jedynie zbiór struktur typu file dla danego procesu, a nie wiedząc nic o liczbie deskryptorów plików otwartych przez ten proces.

Zadanie A1

Załóżmy, że pewien proces, który **nie tworzył procesów potomnych**, korzysta z ***n*** otwartych plików.
Wszystkie te **pliki są różne** (odpowiadają im różne i-węzły).
Załóżmy również, że ten proces jest **jedynym procesem w systemie, który ma otwarte pliki**.

Opisz, jak można taką sytuację wykryć, znając jedynie zbiór struktur typu file dla danego procesu, a nie wiedząc nic o liczbie deskryptorów plików otwartych przez ten proces.

Należy sprawdzić pole f_count w strukturze file. Jeśli jego wartość jest większa od 1, to znaczy, że istnieje więcej deskryptorów niż otwartych plików.

Zadanie A2

Proces P działa według następującego schematu (sześcioelementowa tablica plik zawiera nazwy ścieżkowe plików):

Zadanie A2

Proces P działa według następującego schematu (sześcioelementowa tablica plik zawiera nazwy ścieżkowe plików):

```
fd = open(plik[0], ...);
for (int i = 1; i ≤ 5; ++i)
{
    if (!fork()) break;
    fd = open(plik[i], ...);
}
```

Zadanie A2

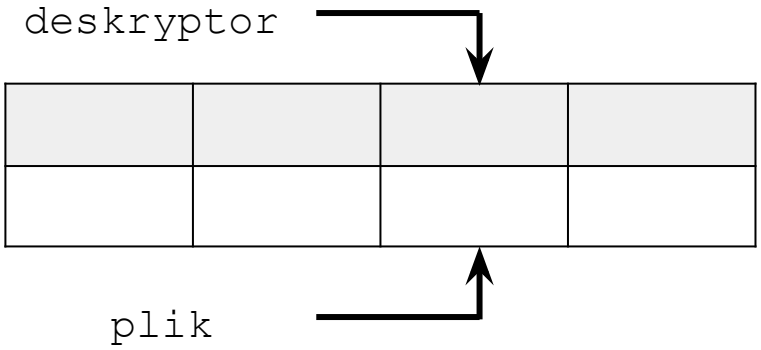
Proces P działa według następującego schematu (sześcioelementowa tablica plik zawiera nazwy ścieżkowe plików):

```
fd = open(plik[0], ...);
for (int i = 1; i ≤ 5; ++i)
{
    if (!fork()) break;
    fd = open(plik[i], ...);
}
```

Jak będą wyglądały tablice deskryptorów plików procesu P i jego potomków?

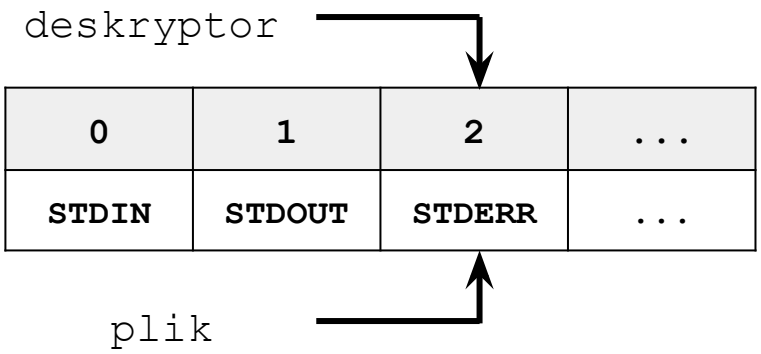
Zadanie A2

Proces rodzic



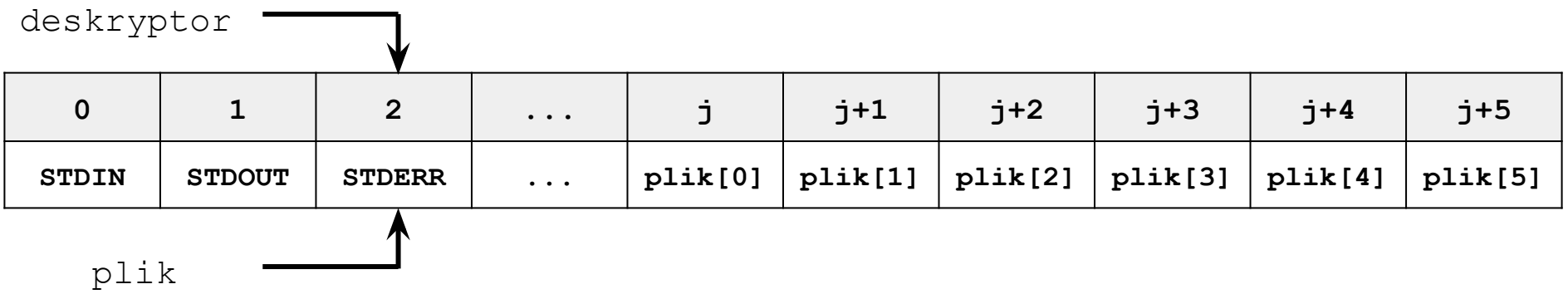
Zadanie A2

Proces rodzic



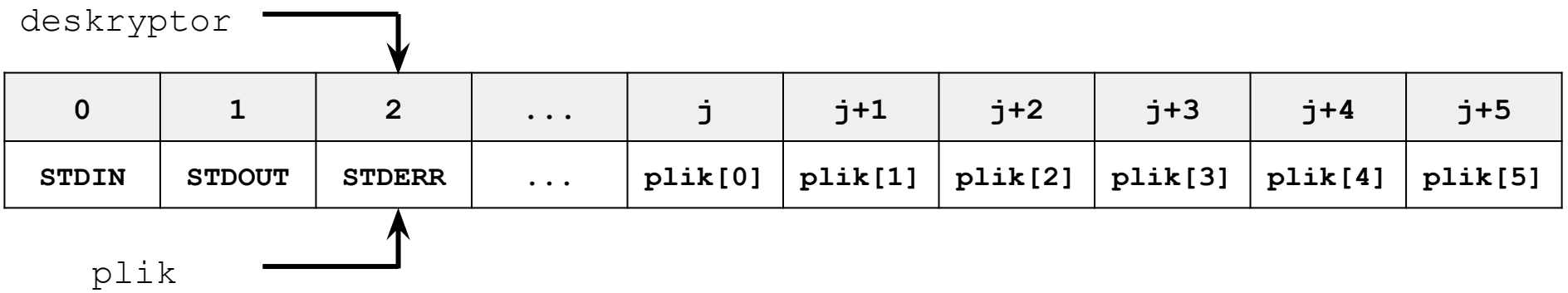
Zadanie A2

Proces rodzic



Zadanie A2

Proces rodzic

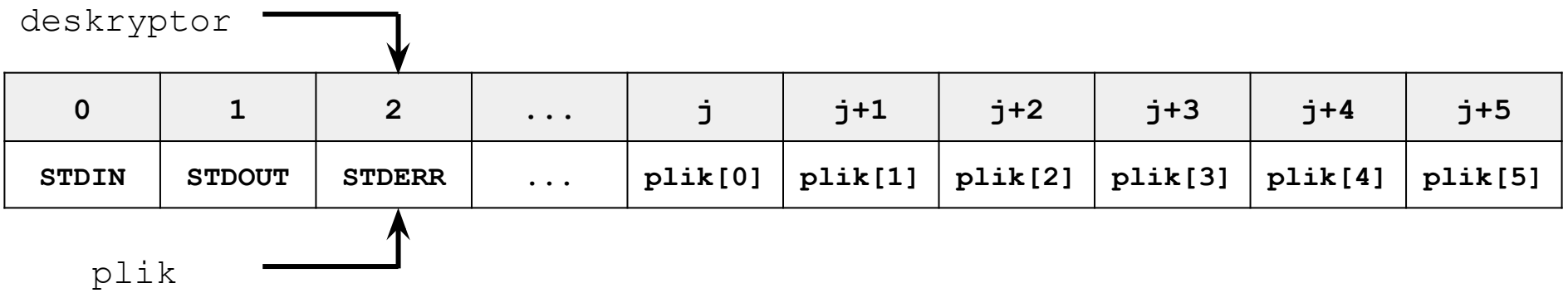


Proces potomny nr 0

0	1	2	...	j
STDIN	STDOUT	STDERR	...	plik[0]

Zadanie A2

Proces rodzic



Proces potomny nr 0

0	1	2	...	j
STDIN	STDOUT	STDERR	...	plik[0]

Proces potomny nr *k*

0	1	2	...	j	...	j+k
STDIN	STDOUT	STDERR	...	plik[0]	...	plik[k]

Zadanie A2

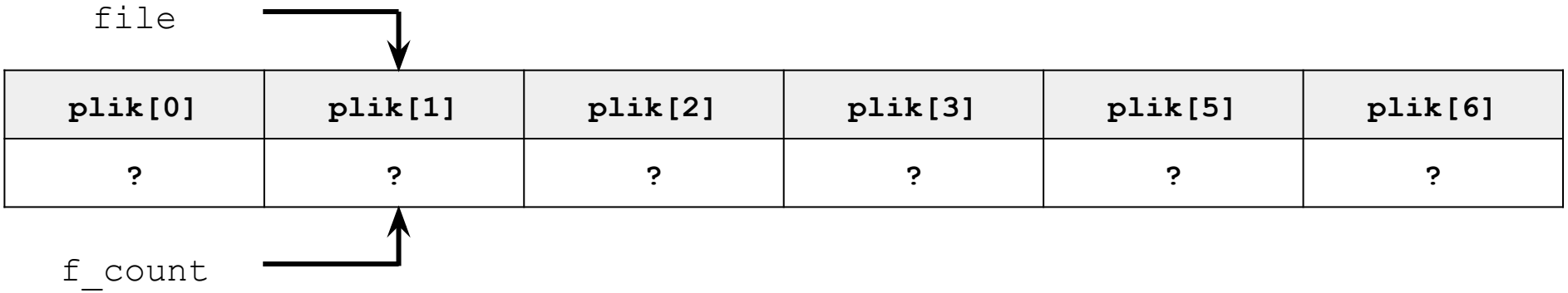
Proces P działa według następującego schematu (sześcioelementowa tablica plik zawiera nazwy ścieżkowe plików):

```
fd = open(plik[0], ...);
for (int i = 1; i ≤ 5; ++i)
{
    if (!fork()) break;
    fd = open(plik[i], ...);
}
```

Jak będzie wyglądała systemowa tablica otwartych plików (jakie wartości będą miały pola f_count)?

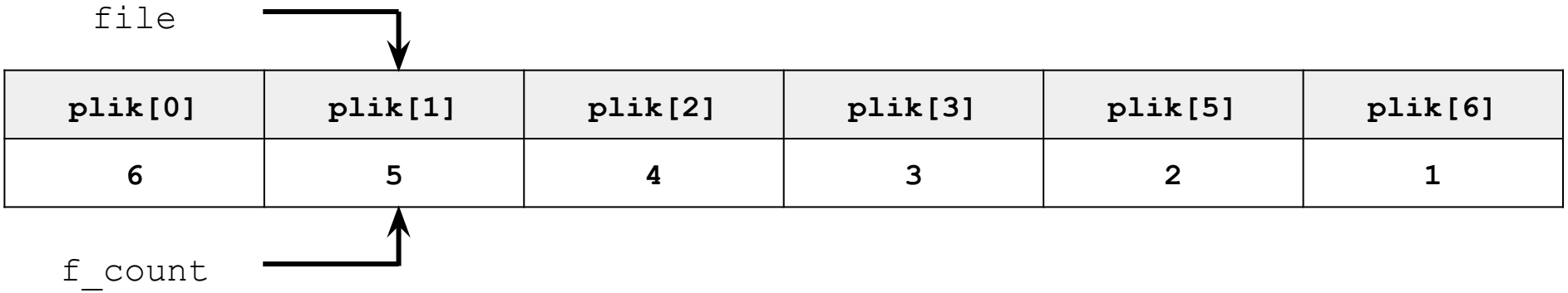
Zadanie A2

Tablica systemowa



Zadanie A2

Tablica systemowa



Filesystem

A filesystem is

http://www.tldp.org/LDP/sag/html/filesystems.html

Filesystem

A filesystem is

the **methods** and **data structures**

http://www.tldp.org/LDP/sag/html/filesystems.html

Filesystem

A filesystem is

the **methods** and **data structures**

that an **operating system** uses to

http://www.tldp.org/LDP/sag/html/filesystems.html

Filesystem

A filesystem is

the **methods** and **data structures**

that an **operating system** uses to

keep track of **files** on a disk or partition

http://www.tldp.org/LDP/sag/html/filesystems.html

Filesystem

A filesystem is

the **methods** and **data structures**

that an **operating system** uses to

keep track of **files** on a disk or partition

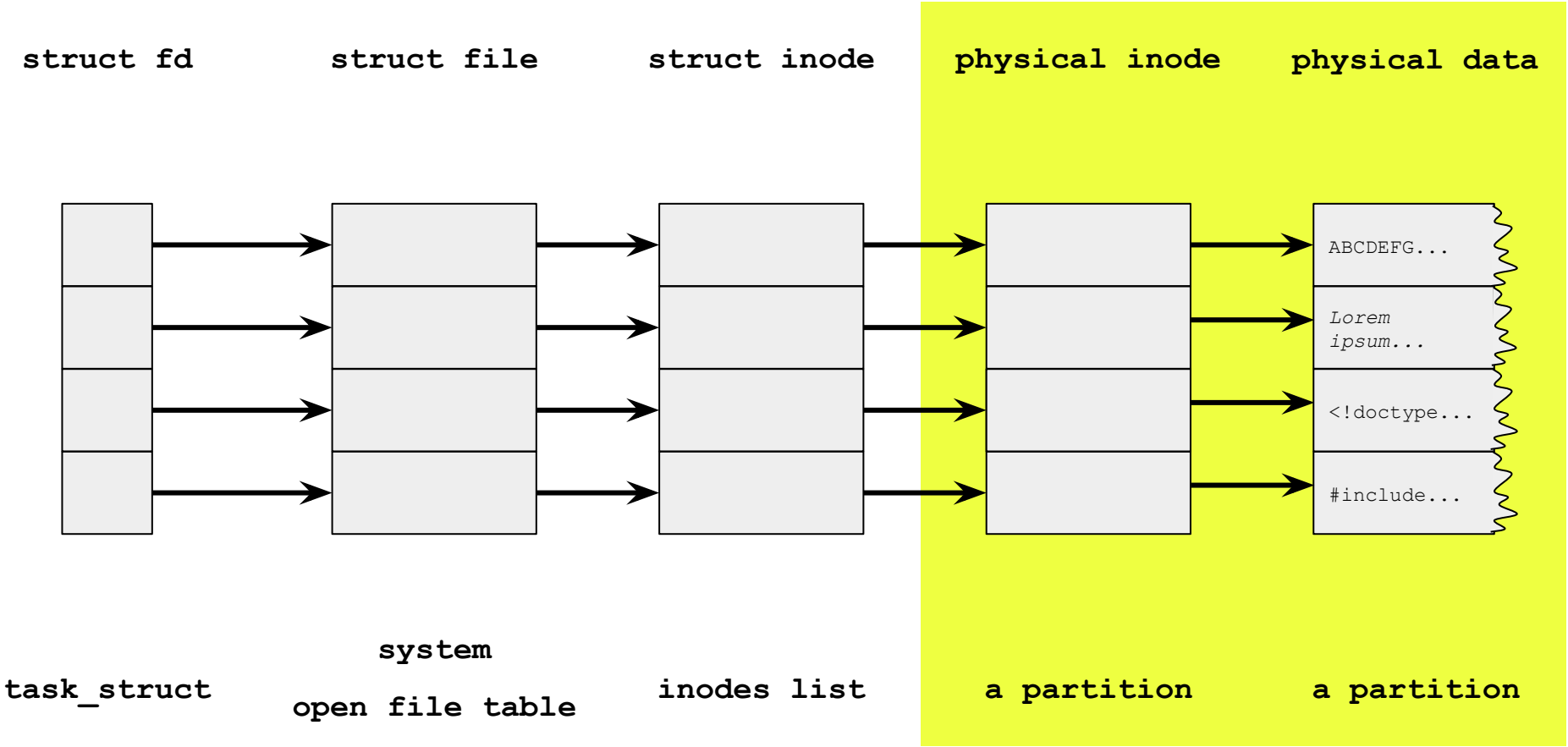
that is: **the way the files are organized on the disk.**

http://www.tldp.org/LDP/sag/html/filesystems.html

Filesystem metadata

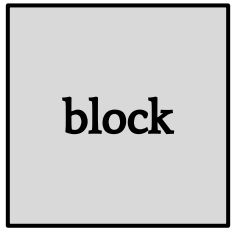
Filesystem metadata

Inodes keep information about files.

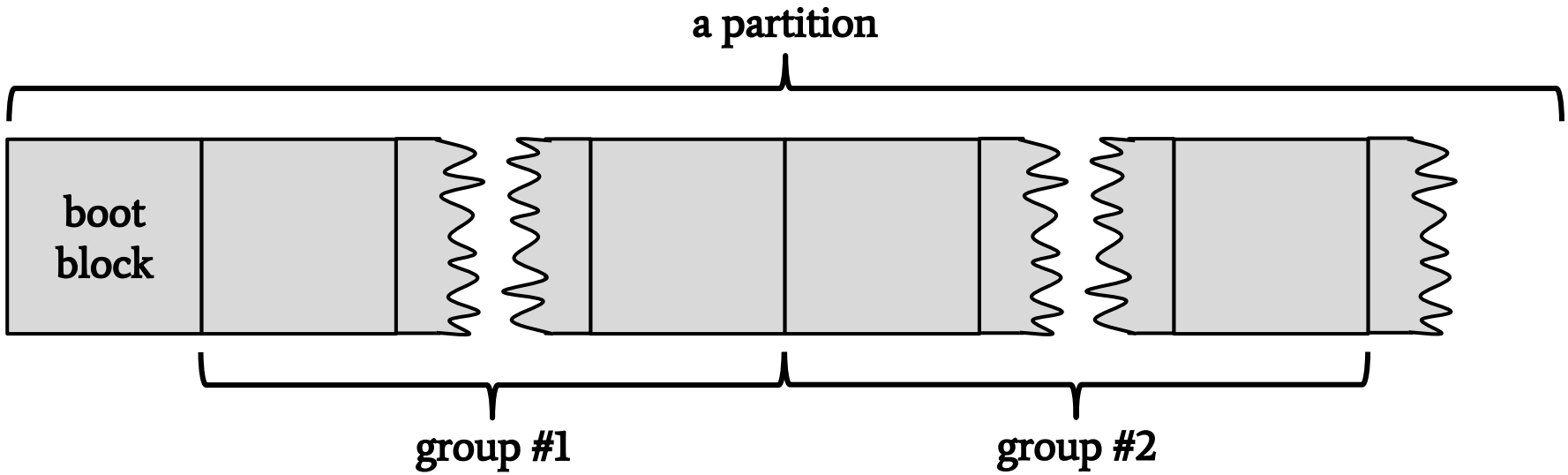


Filesystem metadata

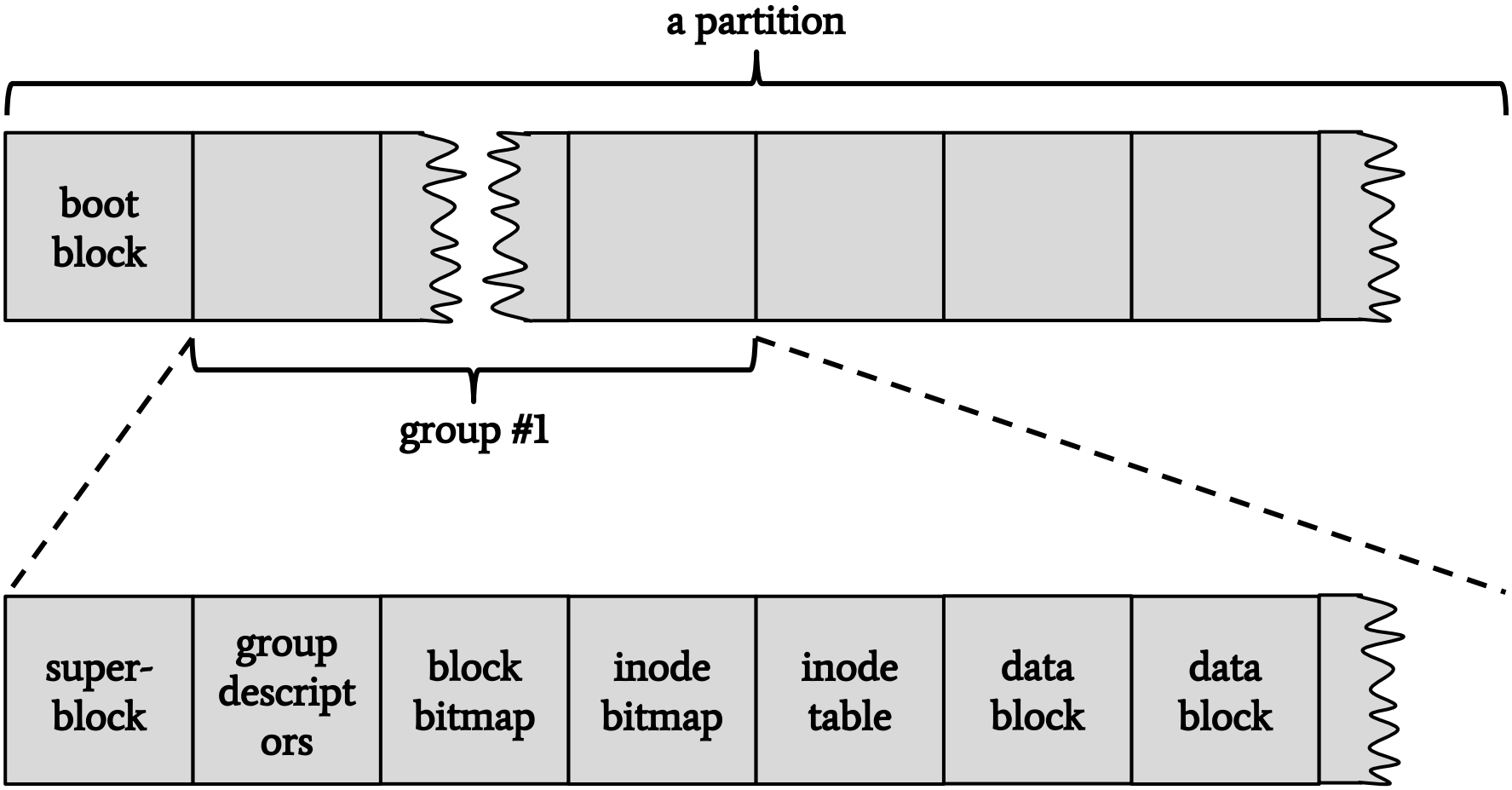
The basic unit:



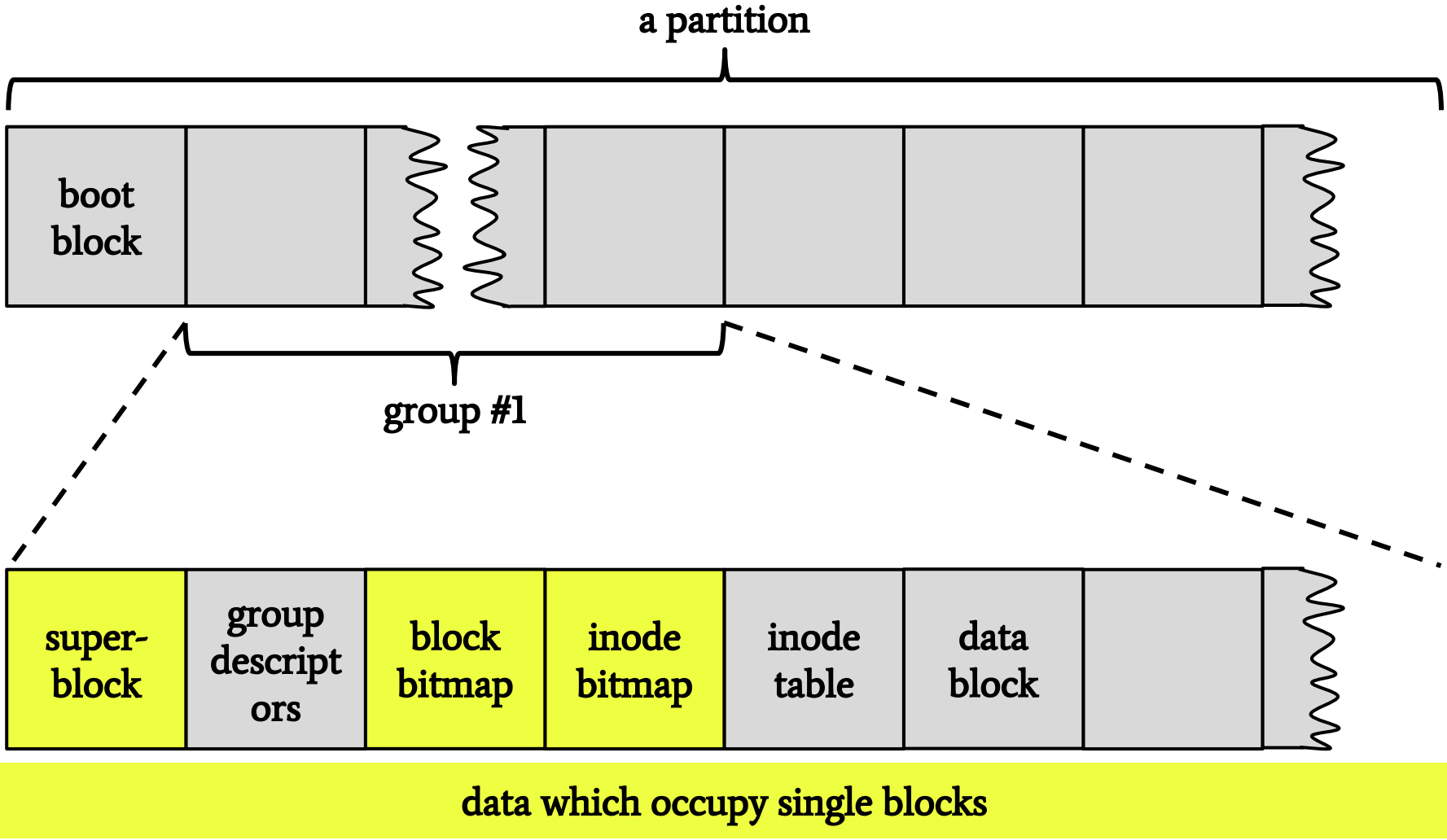
The logical structure of a partition:



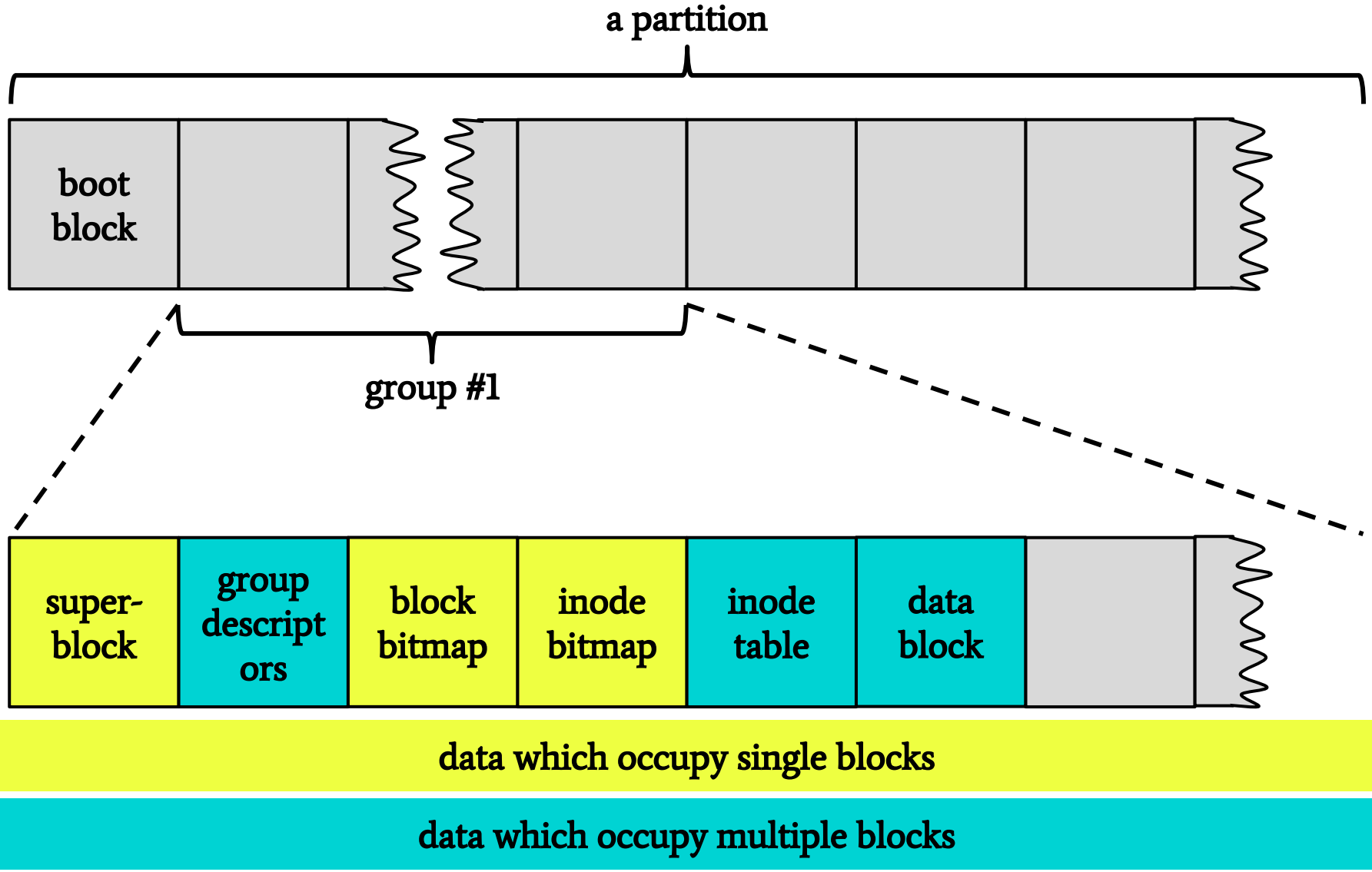
Filesystem metadata



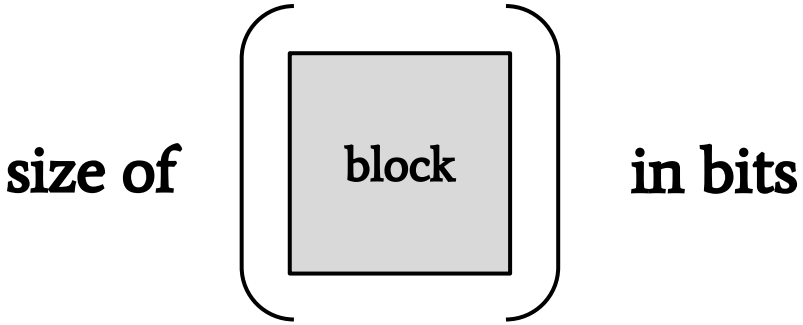
Filesystem metadata



Filesystem metadata



Filesystem metadata



is the **maximal number** of

all blocks and **inodes** in a group

Zadanie B1

Pewna partycja ext2 ma wielkość **4 GiB.**

Rozmiar bloku tej partycji został ustalony na **4 KiB.**

Załóżmy, że:

- rozmiar deskryptora grupy wynosi **48 B,**
- rozmiar i-węzła na dysku wynosi **128 B,**
- a na grupę przypada **4096 i-węzłów.**

Zadanie B1

rozmiar partycji ext2 - **4 GiB**

rozmiar bloku - **4 KiB**

rozmiar deskryptora grupy - **48 B**

rozmiar i-węzła na dysku - **128 B**

na grupę przypada **4096 i-węzłów**

Zadanie B1

rozmiar partycji ext2 – **4 GiB**

rozmiar bloku – **4 KiB**

rozmiar deskryptora grupy – **48 B**

rozmiar i-węzła na dysku – **128 B**

na grupę przypada **4096 i-węzłów**

Jaki jest rozmiar mapy bitowej zajętości bloków?

Zadanie B1

rozmiar partycji ext2 - **4 GiB**

rozmiar bloku - **4 KiB**

rozmiar deskryptora grupy - **48 B**

rozmiar i-węzła na dysku - **128 B**

na grupę przypada **4096 i-węzłów**

Jaki jest rozmiar mapy bitowej zajętości bloków?

4 KiB - czyli rozmiar pojedynczego bloku

Zadanie B1

rozmiar partycji ext2 - **4 GiB**

rozmiar bloku - **4 KiB**

rozmiar deskryptora grupy - **48 B**

rozmiar i-węzła na dysku - **128 B**

na grupę przypada **4096 i-węzłów**

Ile bloków z danymi będzie zawierać każda z grup?

Zadanie B1

rozmiar partycji ext2 - **4 GiB**

rozmiar bloku - **4 KiB**

rozmiar deskryptora grupy - **48 B**

rozmiar i-węzła na dysku - **128 B**

na grupę przypada **4096 i-węzłów**

liczba bloków - **32 * 1024**

Ile bloków z danymi będzie zawierać każda z grup?

Tyle, ile bitów zawiera mapa bitowa zajętości bloków:

4 KiB * 8 b/B = 32 Kib = **32768** b

Zadanie B1

rozmiar partycji ext2 - **4 GiB**

rozmiar bloku - **4 KiB**

rozmiar deskryptora grupy - **48 B**

rozmiar i-węzła na dysku - **128 B**

na grupę przypada **4096 i-węzłów**

liczba bloków - **32 * 1024**

Jaki jest rozmiar danych przechowywanych w jednej grupie bloków?

Zadanie B1

rozmiar partycji ext2 - **4 GiB**

rozmiar bloku - **4 KiB**

rozmiar deskryptora grupy - **48 B**

rozmiar i-węzła na dysku - **128 B**

na grupę przypada **4096 i-węzłów**

liczba bloków - **32 * 1024**

rozmiar grupy - **128 MiB**

Jaki jest rozmiar danych przechowywanych w jednej grupie bloków?

Liczba bloków * rozmiar bloku:

32 * 1024 * 4 KiB = 128 MiB

Zadanie B1

rozmiar partycji ext2 - **4 GiB**

rozmiar bloku - **4 KiB**

rozmiar deskryptora grupy - **48 B**

rozmiar i-węzła na dysku - **128 B**

na grupę przypada **4096 i-węzłów**

liczba bloków - **32 * 1024**

rozmiar grupy - **128 MiB**

Ile grup bloków zostanie utworzonych?

Zadanie B1

rozmiar partycji ext2 - **4 GiB**

rozmiar bloku - **4 KiB**

rozmiar deskryptora grupy - **48 B**

rozmiar i-węzła na dysku - **128 B**

na grupę przypada **4096 i-węzłów**

liczba bloków - **32 * 1024**

rozmiar grupy - **128 MiB**

liczba grup - **32**

Ile grup bloków zostanie utworzonych?

Rozmiar partycji / rozmiar grupy:

$4\ \text{GiB} / 128\ \text{MiB} = (4 * 2^{30})\text{B} / (128 * 2^{20})\text{B} = 4096 / 128 = 32$

Zadanie B1

rozmiar partycji ext2 - **4 GiB**

rozmiar bloku - **4 KiB**

rozmiar deskryptora grupy - **48 B**

rozmiar i-węzła na dysku - **128 B**

na grupę przypada **4096 i-węzłów**

liczba bloków - **32 * 1024**

rozmiar grupy - **128 MiB**

liczba grup - **32**

Ile procent przestrzeni dyskowej zostanie zużyte na metadane?

Zadanie B1

rozmiar partycji ext2 - **4 GiB**

rozmiar bloku - **4 KiB**

rozmiar deskryptora grupy - **48 B**

rozmiar i-węzła na dysku - **128 B**

na grupę przypada **4096 i-węzłów**

liczba bloków - **32 * 1024**

rozmiar grupy - **128 MiB**

liczba grup - **32**

Ile procent przestrzeni dyskowej zostanie zużyte na metadane?

kopia superbloku = 4 KiB
kopia deskryptorów grup = 4 KiB (48 B * 32 = 1536 B < 4 KiB)
mapa bitowa bloków = 4 KiB
mapa bitowa i-węzłów = 4 KiB
tablica i-węzłów = 512 KiB (128 B * 4096 = 512 KiB)

per grupa

w sumie - na 128 MiB, 528 KiB (0,4%) to metadane

// Zaokrąglamy do pełnej liczby bloków.

Zadanie B2

rozmiar partycji - **8 GiB**

rozmiar bloku - **4 KiB**

rozmiar deskryptora grupy - **32 B**

rozmiar i-węzła na dysku - **128 B**

Jak należy skonfigurować system plików na tej partycji, żeby metadane nie zajęły więcej niż 2% partycji?

- Ile będzie grup dyskowych?
- Ile bloków zajmie jedna grupa?
- Co będzie się znajdowało w kolejnych blokach dyskowych partycji?
- Jakie metadane będą przechowywane w kolejnych (ilu) blokach?
- Ile bloków będzie przypadało średnio na jeden plik?

Blok startowy można zaniedbać w rozważaniach.

Zadanie B2

rozmiar partycji - **8 GiB**

rozmiar bloku - **4 KiB**

rozmiar deskryptora grupy - **32 B**

rozmiar i-węzła na dysku - **128 B**

liczba grup -

bloki w grupie -

bloki na plik -

Jak należy skonfigurować system plików na tej partycji, żeby metadane nie zajęły więcej niż 2% partycji?

- Ile będzie grup dyskowych?
- Ile bloków zajmie jedna grupa?
- Co będzie się znajdowało w kolejnych blokach dyskowych partycji?
- Jakie metadane będą przechowywane w kolejnych (ilu) blokach?
- Ile bloków będzie przypadało średnio na jeden plik?

Blok startowy można zaniedbać w rozważaniach.

Zadanie B2

rozmiar partycji - **8 GiB**

rozmiar bloku - **4 KiB**

rozmiar deskryptora grupy - **32 B**

rozmiar i-węzła na dysku - **128 B**

liczba grup -

bloki w grupie -

bloki na plik -

Jak należy skonfigurować system plików na tej partycji, żeby metadane nie zajęły więcej niż 2% partycji?

Zadanie B2

rozmiar partycji - **8 GiB**

rozmiar bloku - **4 KiB**

rozmiar deskryptora grupy - **32 B**

rozmiar i-węzła na dysku - **128 B**

liczba grup -

bloki w grupie -

bloki na plik -

Jak należy skonfigurować system plików na tej partycji, żeby metadane nie zajęły więcej niż 2% partycji?

bloki w grupie = **$32 * 2^{10}$**

rozmiar grupy = $32 * 2^{10} * 4 * 2^{10}$ B = $128 * 2^{20}$ B = **128 MiB**

liczba grup = $8 * 2^{30} / 128 * 2^{20} = 8 * 2^{10} / 128$ = **64**

liczba bloków tabeli i-węzłów = **x**

$64 * (4 * 2^{10}$ B + [64 * 32 B] + $4 * 2^{10} + 4 * 2^{10} + x * 4 * 2^{10}$)

// Zaokrąglamy do pełnej liczby bloków.

Zadanie B2

rozmiar partycji - **8 GiB**

rozmiar bloku - **4 KiB**

rozmiar deskryptora grupy - **32 B**

rozmiar i-węzła na dysku - **128 B**

liczba grup -

bloki w grupie -

bloki na plik -

Jak należy skonfigurować system plików na tej partycji, żeby metadane nie zajęły więcej niż 2% partycji?

bloki w grupie = **$32 * 2^{10}$**

rozmiar grupy = $32 * 2^{10} * 4 * 2^{10}$ B = $128 * 2^{20}$ B = **128 MiB**

liczba grup = $8 * 2^{30} / 128 * 2^{20} = 8 * 2^{10} / 128$ = **64**

liczba bloków tabeli i-węzłów = **x**

$$\mathbf{64 * (4 * 2^{10}\ B + 4 * 2^{10} + 4 * 2^{10} + 4 * 2^{10} + x * 4 * 2^{10})}$$

$$\mathbf{< 8 * 2^{30} * 0.02\ B}$$

Zadanie B2

rozmiar partycji - **8 GiB**

rozmiar bloku - **4 KiB**

rozmiar deskryptora grupy - **32 B**

rozmiar i-węzła na dysku - **128 B**

liczba grup -

bloki w grupie -

bloki na plik -

Jak należy skonfigurować system plików na tej partycji, żeby metadane nie zajęły więcej niż 2% partycji?

bloki w grupie = **$32 * 2^{10}$**

rozmiar grupy = $32 * 2^{10} * 4 * 2^{10}$ B = $128 * 2^{20}$ B = **128 MiB**

liczba grup = $8 * 2^{30} / 128 * 2^{20} = 8 * 2^{10} / 128$ = **64**

liczba bloków tabeli i-węzłów = **x**

$64 * 4 * 2^{10} * (4 + x)$ B < $167772 * 2^{10}$ B

Zadanie B2

rozmiar partycji - **8 GiB**

rozmiar bloku - **4 KiB**

rozmiar deskryptora grupy - **32 B**

rozmiar i-węzła na dysku - **128 B**

liczba grup -

bloki w grupie -

bloki na plik -

Jak należy skonfigurować system plików na tej partycji, żeby metadane nie zajęły więcej niż 2% partycji?

bloki w grupie = **$32 * 2^{10}$**

rozmiar grupy = $32 * 2^{10} * 4 * 2^{10}$ B = $128 * 2^{20}$ B = **128 MiB**

liczba grup = $8 * 2^{30} / 128 * 2^{20} = 8 * 2^{10} / 128$ = **64**

liczba bloków tabeli i-węzłów = **x**

$x \leq 650$

Zadanie B2

rozmiar partycji - **8 GiB**

rozmiar bloku - **4 KiB**

rozmiar deskryptora grupy - **32 B**

rozmiar i-węzła na dysku - **128 B**

liczba grup -

bloki w grupie -

bloki na plik -

Jak należy skonfigurować system plików na tej partycji, żeby metadane nie zajęły więcej niż 2% partycji?

bloki w grupie = **$32 * 2^{10}$**

rozmiar grupy = $32 * 2^{10} * 4 * 2^{10}$ B = $128 * 2^{20}$ B = **128 MiB**

liczba grup = $8 * 2^{30} / 128 * 2^{20} = 8 * 2^{10} / 128$ = **64**

liczba bloków tabeli i-węzłów = **650**

limit liczby i-węzłów = (650 * 4 * 2^{10} B / 128 B) czy 8 * 4 * 2^{10} ?

Zadanie B2

rozmiar partycji - **8 GiB**

rozmiar bloku - **4 KiB**

rozmiar deskryptora grupy - **32 B**

rozmiar i-węzła na dysku - **128 B**

liczba grup -

bloki w grupie -

bloki na plik -

Jak należy skonfigurować system plików na tej partycji, żeby metadane nie zajęły więcej niż 2% partycji?

bloki w grupie = **$32 * 2^{10}$**

rozmiar grupy = $32 * 2^{10} * 4 * 2^{10}$ B = $128 * 2^{20}$ B = **128 MiB**

liczba grup = $8 * 2^{30} / 128 * 2^{20} = 8 * 2^{10} / 128$ = **64**

liczba bloków tabeli i-węzłów = **650**

limit liczby i-węzłów = $(650 * 4 * 2^{10}$ B / 128 B) < $8 * 4 * 2^{10}$

Zadanie B2

rozmiar partycji - **8 GiB**

rozmiar bloku - **4 KiB**

rozmiar deskryptora grupy - **32 B**

rozmiar i-węzła na dysku - **128 B**

liczba grup -

bloki w grupie -

bloki na plik -

Jak należy skonfigurować system plików na tej partycji, żeby metadane nie zajęły więcej niż 2% partycji?

bloki w grupie = **$32 * 2^{10}$**

rozmiar grupy = $32 * 2^{10} * 4 * 2^{10}$ B = $128 * 2^{20}$ B = **128 MiB**

liczba grup = $8 * 2^{30} / 128 * 2^{20} = 8 * 2^{10} / 128$ = **64**

liczba bloków tabeli i-węzłów = **650**

limit liczby i-węzłów = **20800**

Zadanie B2

rozmiar partycji - **8 GiB**

rozmiar bloku - **4 KiB**

rozmiar deskryptora grupy - **32 B**

rozmiar i-węzła na dysku - **128 B**

liczba grup -

bloki w grupie -

bloki na plik -

Jak należy skonfigurować system plików na tej partycji, żeby metadane nie zajęły więcej niż 2% partycji?

bloki w grupie = **$32 * 2^{10}$**

rozmiar grupy = $32 * 2^{10} * 4 * 2^{10}$ B = $128 * 2^{20}$ B = **128 MiB**

liczba grup = $8 * 2^{30} / 128 * 2^{20} = 8 * 2^{10} / 128$ = **64**

liczba bloków tabeli i-węzłów = **650**

limit liczby i-węzłów = **20800**

liczba bloków na plik = liczba bloków z plikami/liczba i-węzłów

Zadanie B2

rozmiar partycji - **8 GiB**

rozmiar bloku - **4 KiB**

rozmiar deskryptora grupy - **32 B**

rozmiar i-węzła na dysku - **128 B**

liczba grup -

bloki w grupie -

bloki na plik -

Jak należy skonfigurować system plików na tej partycji, żeby metadane nie zajęły więcej niż 2% partycji?

bloki w grupie = **$32 * 2^{10}$**

rozmiar grupy = $32 * 2^{10} * 4 * 2^{10}$ B = $128 * 2^{20}$ B = **128 MiB**

liczba grup = $8 * 2^{30} / 128 * 2^{20} = 8 * 2^{10} / 128$ = **64**

liczba bloków tabeli i-węzłów = **650**

limit liczby i-węzłów = **20800**

liczba bloków na plik = $(8 * 4 * 2^{10} - 4 - 650)$/liczba i-węzłów

Zadanie B2

rozmiar partycji - **8 GiB**

rozmiar bloku - **4 KiB**

rozmiar deskryptora grupy - **32 B**

rozmiar i-węzła na dysku - **128 B**

liczba grup -

bloki w grupie -

bloki na plik -

Jak należy skonfigurować system plików na tej partycji, żeby metadane nie zajęły więcej niż 2% partycji?

bloki w grupie = **$32 * 2^{10}$**

rozmiar grupy = $32 * 2^{10} * 4 * 2^{10}$ B = $128 * 2^{20}$ B = **128 MiB**

liczba grup = $8 * 2^{30} / 128 * 2^{20} = 8 * 2^{10} / 128$ = **64**

liczba bloków tabeli i-węzłów = **650**

limit liczby i-węzłów = **20800**

liczba bloków na plik = 32114/20800 ≈ **1,5**

Zadanie B2

rozmiar partycji - **8 GiB**

rozmiar bloku - **4 KiB**

rozmiar deskryptora grupy - **32 B**

rozmiar i-węzła na dysku - **128 B**

liczba grup - **64**

bloki w grupie - **$32 * 2^{10}$**

bloki na plik - **1,5**

Jak należy skonfigurować system plików na tej partycji, żeby metadane nie zajęły więcej niż 2% partycji?

bloki w grupie = **$32 * 2^{10}$**

rozmiar grupy = $32 * 2^{10} * 4 * 2^{10}$ B = $128 * 2^{20}$ B = **128 MiB**

liczba grup = $8 * 2^{30} / 128 * 2^{20} = 8 * 2^{10} / 128$ = **64**

liczba bloków tabeli i-węzłów = **650**

limit liczby i-węzłów = **20800**

liczba bloków na plik = 32114/20800 ≈ **1,5**

Zadanie B3

1. Załóżmy, że blok na dysku ma rozmiar 1024 B. Jaki rozmiar ma grupa?

Zadanie B3

1. Załóżmy, że blok na dysku ma rozmiar 1024 B. Jaki rozmiar ma grupa?

liczba bloków w grupie = liczba bitów w bitmapie

liczba bloków w grupie = 8 * 1024

rozmiar grupy = liczba bloków * rozmiar bloku

rozmiar grupy = 8 * 1024 * 1024 B = 8 * 2^{30} B = **8 MiB**

Zadanie B3

2. Załóżmy, że grupa bloków ma rozmiar 128 MiB. Ile wynosi rozmiar bloku na dysku?

Zadanie B3

2. Załóżmy, że grupa bloków ma rozmiar 128 MiB. Ile wynosi rozmiar bloku na dysku?

rozmiar bloku w bajtach = x

liczba bloków = 8 * x

rozmiar grupy = 8 * x * x = 128 MiB

x^2 = 16 MiB

x = 4 KiB

Zadanie B3

rozmiar bloku - **1 KiB**

rozmiar deskryptora grupy - **24 B**

liczba grup - **400**

3. Który bit w mapie bitowej zajętości bloków w grupie zawierającej wszystkie deskryptory grup odpowiada blokowi zawierającemu tę mapę?

Zadanie B3

rozmiar bloku - **1 KiB**

rozmiar deskryptora grupy - **24 B**

liczba grup - **400**

3. Który bit w mapie bitowej zajętości bloków w grupie zawierającej wszystkie deskryptory grup odpowiada blokowi zawierającemu tę mapę?

liczba bloków zajętych przez deskryptory grup =**N**

N = ⌈400 * 24 B / 1 KiB⌉ = ⌈9600 B / 1024 B⌉ = 10

bitmapę bloków poprzedza 11 bloków (superblok i deskryptory grup)

bitmapie bloków odpowiada 12. bit

Zadanie B3

4. Dlaczego funkcja kontrolująca spójność mapy bitowej dla każdej z tych map zakończy się błędem?

```
0111 1111 1101 1010 0110 ...
```

```
1011 0000 0000 0000 0000 ...
```

```
1101 1111 1111 1111 1111 ...
```

Zadanie B3

4. Dlaczego funkcja kontrolująca spójność mapy bitowej dla każdej z tych map zakończy się błędem?

0111 1111 1101 1010 0110 ...

Pierwszy blok jest zawsze zajęty - mieści się tam superblok.

1011 0000 0000 0000 0000 ...

1101 1111 1111 1111 1111 ...

Zadanie B3

4. Dlaczego funkcja kontrolująca spójność mapy bitowej dla każdej z tych map zakończy się błędem?

0111 1111 1101 1010 0110 ...

Pierwszy blok jest zawsze zajęty - mieści się tam superblok.

1011 0000 0000 0000 0000 ...

Drugi blok jest zawsze zajęty - mamy co najmniej jedną grupę, więc potrzebujemy zapisać co najmniej jeden deskryptor grupy.

1101 1111 1111 1111 1111 ...

Zadanie B3

4. Dlaczego funkcja kontrolująca spójność mapy bitowej dla każdej z tych map zakończy się błędem?

0111 1111 1101 1010 0110 ...

Pierwszy blok jest zawsze zajęty - mieści się tam superblok.

1011 0000 0000 0000 0000 ...

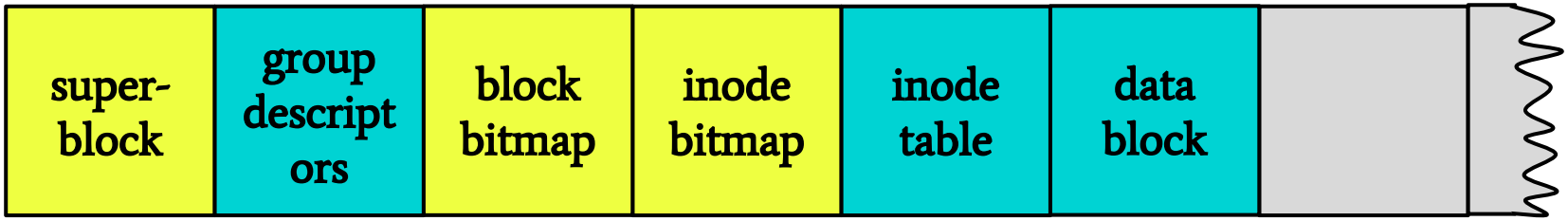
Drugi blok jest zawsze zajęty - mamy co najmniej jedną grupę, więc potrzebujemy zapisać co najmniej jeden deskryptor grupy.

1101 1111 1111 1111 1111 ...

Trzeci blok mieści albo dalsze deskryptory grup, albo bitmapę bloków - jest więc zawsze zajęty.

Recap

A single group contains:

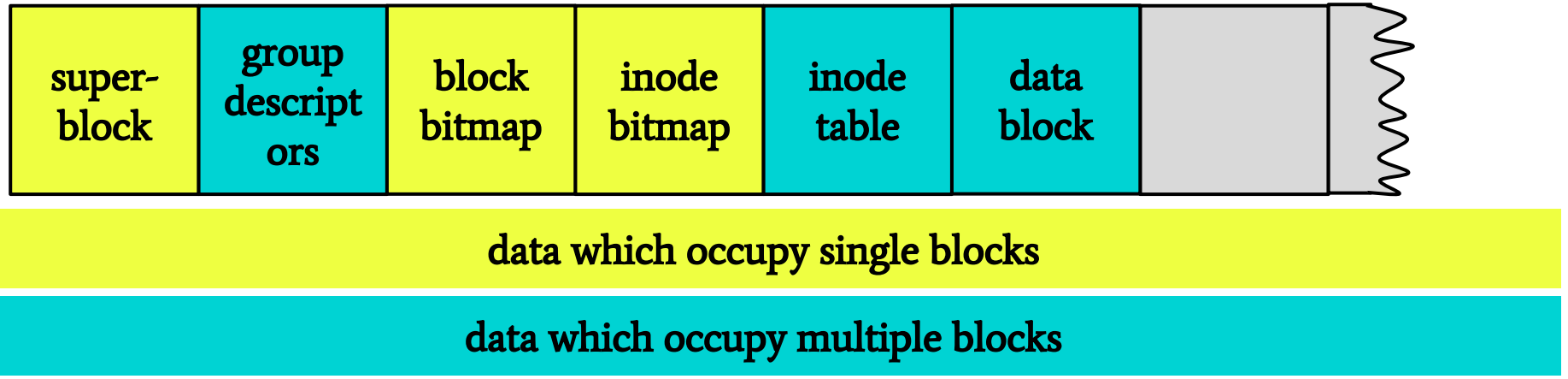


data which occupy single blocks

data which occupy multiple blocks

Recap

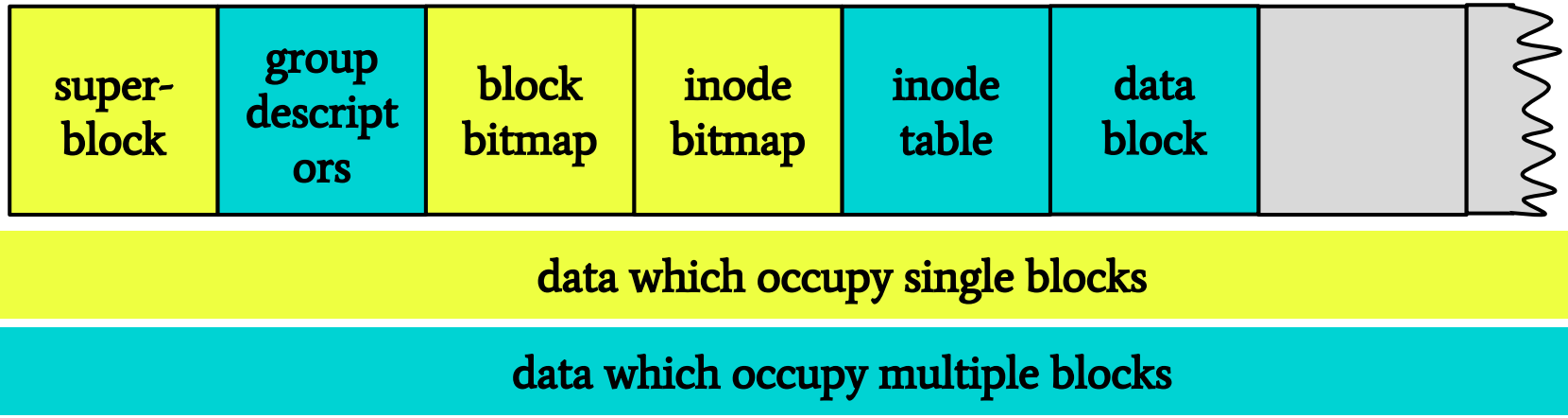
A single group contains:



If we know the block size we can compute the size of a group.

Recap

A single group contains:



If we know the block size we can compute the size of a group.

It is true because...

...,maximal number of blocks in a group
is determined by
the number of bits in the block bitmap.

Zadanie B4

rozmiar bloku - **4 KiB**

rozmiar deskryptora grupy - **64 B**

Jaki jest maksymalny rozmiar partycji dla systemu plików, w którym pierwsza grupa przechowuje wszystkie deskryptory grup danej partycji?

Zadanie B4

rozmiar bloku - **4 KiB**

rozmiar deskryptora grupy - **64 B**

Jaki jest maksymalny rozmiar partycji dla systemu plików, w którym pierwsza grupa przechowuje wszystkie deskryptory grup danej partycji?

rozmiar grupy = 8 * (4 * 1024) * (4 * 1024) B = 128 MiB = 2^{27} B

➔ 1 blok musimy poświęcić na superblok, a resztę można zużyć na przechowanie deskryptorów grup

➔ zaokrąglamy wynik, pomijając superblok

możliwa liczba grup = możliwa liczba deskryptorów grup = $2^{27} / 2^{6} = 2^{21}$

rozmiar partycji = możliwa liczba grup * rozmiar grupy = $2^{21} * 2^{27}$ B

rozmiar partycji = **256 TiB**

Zadanie B4

rozmiar bloku - **4 KiB**

rozmiar deskryptora grupy - **64 B**

Jaki byłby w takiej sytuacji rozmiar metagrupy?

Zadanie B4

rozmiar bloku - **4 KiB**

rozmiar deskryptora grupy - **64 B**

Jaki byłby w takiej sytuacji rozmiar metagrupy?

Metagrupa zawiera tyle grup, aby ich deskryptory zmieściły się w jednym bloku, dlatego:

liczba grup w metagrupie = $4 * 2^{10}$ B / 64 B = 2^{12} B / 2^{6} B = **64**