

# Systemy operacyjne

Mariusz Żynel

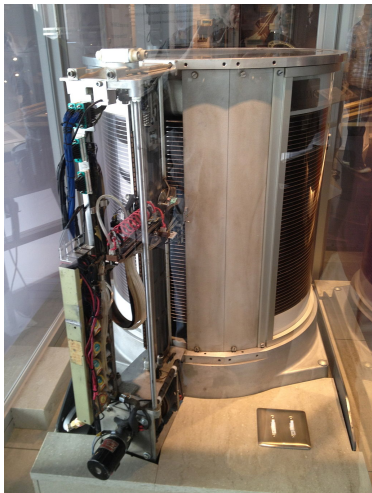
`mariusz@math.uwb.edu.pl`

`http://math.uwb.edu.pl/~mariusz/`

Uniwersytet w Białymstoku

2023/2024

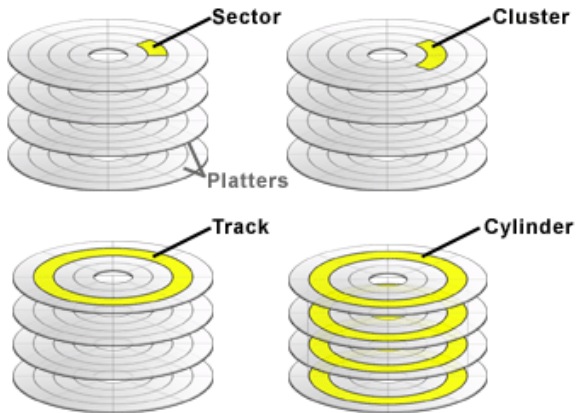
# Dysk twardy HDD - budowa



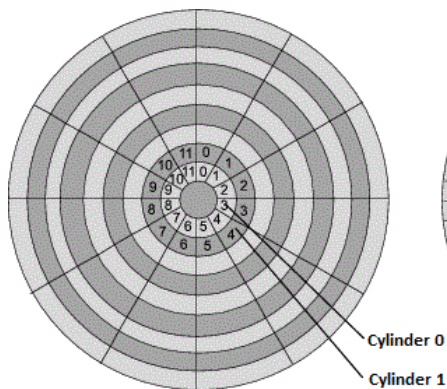
IBM 350 RAMAC (1956)



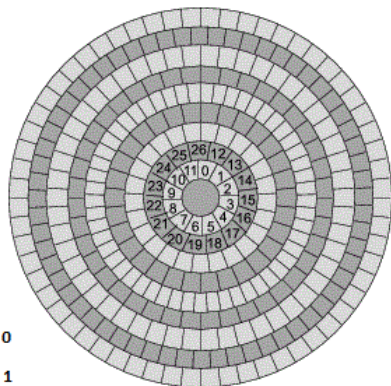
# Dysk twardy HDD - sektory, ścieżki, cylindry



# Dysk twardy HDD - adresowanie



**CHS Addressing**



**LBA Addressing**

## System plików

Sposób przechowywania i organizacji danych na nośniku w formie plików

- UFS to hierarchiczny, dyskowy system plików ogólnego przeznaczenia zaprojektowany przez Sun Microsystems na bazie FFS (Fast File System) z BSD na potrzeby systemu SunOS
- Założenia UFS :
  - Ułożyć pliki na dysku tak, aby dostęp do nich zajmował możliwie mało czasu i tak aby tracić możliwie mało miejsca
  - Przesłać możliwie dużo danych przy każdej operacji I/O
  - Zminimalizować przeszukiwanie (seek time) i opóźnienia (latency)
- Charakterystyka UFS:
  - Wszystkie metadane pliku zgromadzone są w **inode** (index node)
  - Katalog zawiera tylko nazwy plików skojarzone z numerami inode

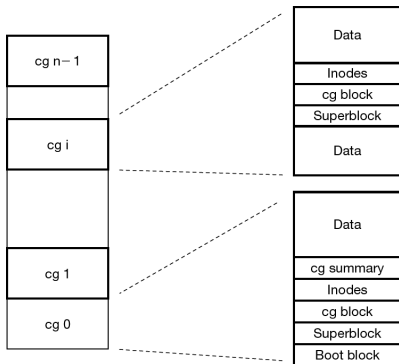
- 1985 (SunOS 2.0) – dostosowanie FFS do vnode/vfs, powstaje UFS
- 1992 (SunOS 4.1) – obsługa plików 1TB
- 1994 (Solaris 2.4) – księgowanie metadanych (metadata logging)
- 1995 (Solaris 2.5) – wprowadzenie ACL
- 2003 (Solaris 9 U4) – obsługa dużych plików > 1TB

# UFS – schemat

**Block** 16 sektorów, 8KB danych, wczytywane/zapisywane w jednej operacji I/O

**Boot block** zawiera kod wykonywany podczas uruchamiania systemu

**Superblock** zawiera dane na temat dysku i systemu plików



**Cylinder group** zgrupowane cylindry dysku, aby zminimalizować czas przeszukiwania dla danych, które powinny być blisko siebie. Blisko siebie powinny być inody, bloki pośrednie i bloki danych katalogów i plików w nich się znajdujących. Podkatalogi mogą być umieszczone w innych grupach cylindrów.

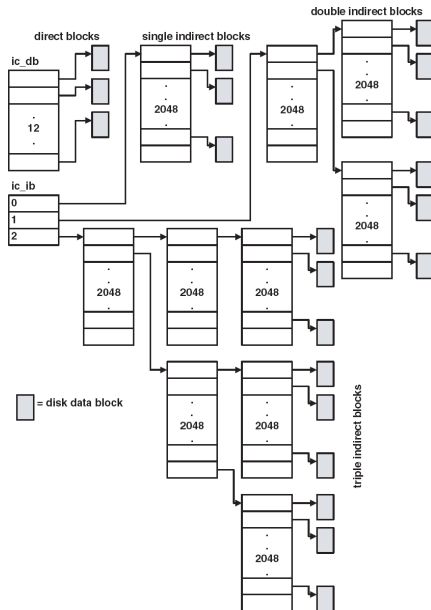
# UFS – inode

```
#define NDADDR 12          /* direct addresses in inode */
#define NIADDR 3          /* indirect addresses in inode */

struct icommon {
    o_mode_t    ic_smode;      /* 0: mode and type of file */
    short       ic_nlink;     /* 2: number of links to file */
    o_uid_t     ic_suid;      /* 4: owner's user id */
    o_gid_t     ic_sgid;      /* 6: owner's group id */
    u_offset_t  ic_lsize;     /* 8: number of bytes in file */
    struct timeval32 ic_atime; /* 16: time last accessed */
    struct timeval32 ic_mtime; /* 24: time last modified */
    struct timeval32 ic_ctime; /* 32: last time inode changed */
    daddr32_t   ic_db[NDADDR]; /* 40: disk block addresses */
    daddr32_t   ic_ib[NIADDR]; /* 88: indirect blocks */
    int32_t     ic_flags;     /* 100: cflags */
    int32_t     ic_blocks;    /* 104: 512 byte blocks actually held */
    int32_t     ic_gen;      /* 108: generation number */
    int32_t     ic_shadow;   /* 112: shadow inode */
    uid_t       ic_uid;      /* 116: long EFT version of uid */
    gid_t       ic_gid;      /* 120: long EFT version of gid */
    uint32_t    ic_oeftflag;  /* 124: extended attr directory ino, 0 = none */
};
```



# UFS – alokacja bloków danych



# UFS – alokacja a rozmiary plików

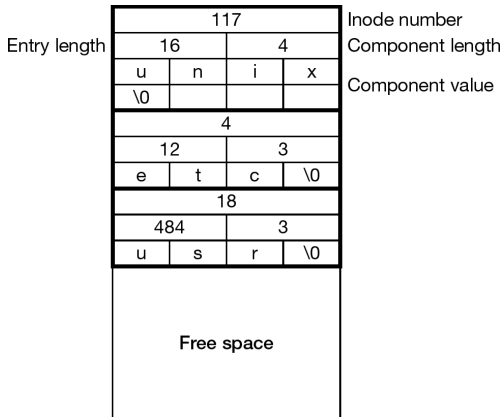
Poziom	Liczba bloków danych	Rozmiar pliku*
Bezpośredni	12	96KB
Pierwszy poziom pośredni	$12 + 2048$	16MB
Drugi poziom pośredni	$12 + 2048 + 2048^2$	32GB
Trzeci poziom pośredni	$12 + 2048 + 2048^2 + 2048^3$	64TB

\* W praktyce maksymalnie 16TB z uwagi na konieczność adresowania fragmentów po 1KB

# UFS – katalog

```
#define MAXNAMLEN 255

struct direct {
    uint32_t    d_ino;           /* inode number of entry */
    ushort_t   d_reclen;       /* length of this record */
    ushort_t   d_namlen;      /* length of string in d_name */
    char       d_name[MAXNAMLEN + 1]; /* name must be no longer than this */
};
```



# Dowiązania plików

- Twarde linki (hard links)
  - Twardy link to bezpośrednie odwołanie do inode danego pliku, albo w tym samym katalogu poprzez inną nazwę, albo w innym katalogu
  - Mogą być tworzone tylko do zwykłych plików
  - Mogą być tworzone w obrębie jednego systemu plików
  - Jako pliki są od siebie nieodróżnialne, tylko ścieżka jest inna

```
ln plik1 plik2
```

```
ln /etc/init.d/apache /etc/rc3.d/S50apache
```

- Linki symboliczne (symbolic, soft links)
  - Link symboliczny to pośrednie odniesienie do pliku, za pośrednictwem specjalnego pliku typu IFLNK, który ma własny inode i jako dane zawiera ścieżkę do danego pliku
  - Mogą odnosić się do katalogów
  - Mogą odnosić się do innych systemów plików
  - Prawa dostępu do linku i pliku docelowego mogą się różnić (`ls -lL`)

```
ln -s /opt/cfw/sbin/sendmail /usr/sbin/sendmail
```

```
ln -s dokumenty/*.txt .
```

# Tworzenie i montowanie systemu plików

`fdisk` zarządzanie partycjami MBR na dysku

`format` zarządzanie partycjami Solaris

`newfs` konstruowanie nowego systemu plików UFS

```
newfs /dev/rdisk/c1t3d0s5
```

`mount` montowanie systemu plików

```
mount /dev/dsk/c1t3d0s5 /opt
```

`umount` odmontowanie systemu plików

```
umount /opt
```

`/etc/vfstab` spis parametrów montowania systemów plików

# Konserwacja systemu plików

`df` zwraca informacje o zamontowanych systemach plików

```
df -h -F ufs
```

`fsck` sprawdzenie spójności i ewentualna naprawa systemu plików

```
fsck /dev/rdisk/c1t3d0s5
```

`ufsdump` archiwizacja systemu plików UFS

```
ufsdump 0f /backup/sys/opt.fs /dev/rdisk/c1t3d0s5
```

`ufsrestore` odtwarzanie systemu plików UFS

```
ufsrestore rf /mnt/backup/system/opt.fs
```

- ZFS – Zettabyte File System
- NFS – Network File System
- FAT – File Allocation Table
- NTFS – NT File System
- ext, ext2, ext3, ext4 – extended file system
- ReiserFS – Reiser File System