

Systemy operacyjne

WYKŁAD 5

dr inż. Stanisława Plichta
splichta@ans-ns.edu.pl

Struktura extentu

```
#define EXT4_MIN_BLOCK_SIZE 1024
#define EXT4_MAX_BLOCK_SIZE 65536 /* oczywiście bajtów */

struct ext4_extent {
    __le32 ee_block;        /* first logical block extent covers */
    __le16 ee_len;          /* number of blocks covered by extent */
    __le16 ee_start_hi;     /* high 16 bits of physical block */
    __le32 ee_start_lo;    /* low 32 bits of physical block */
};
```

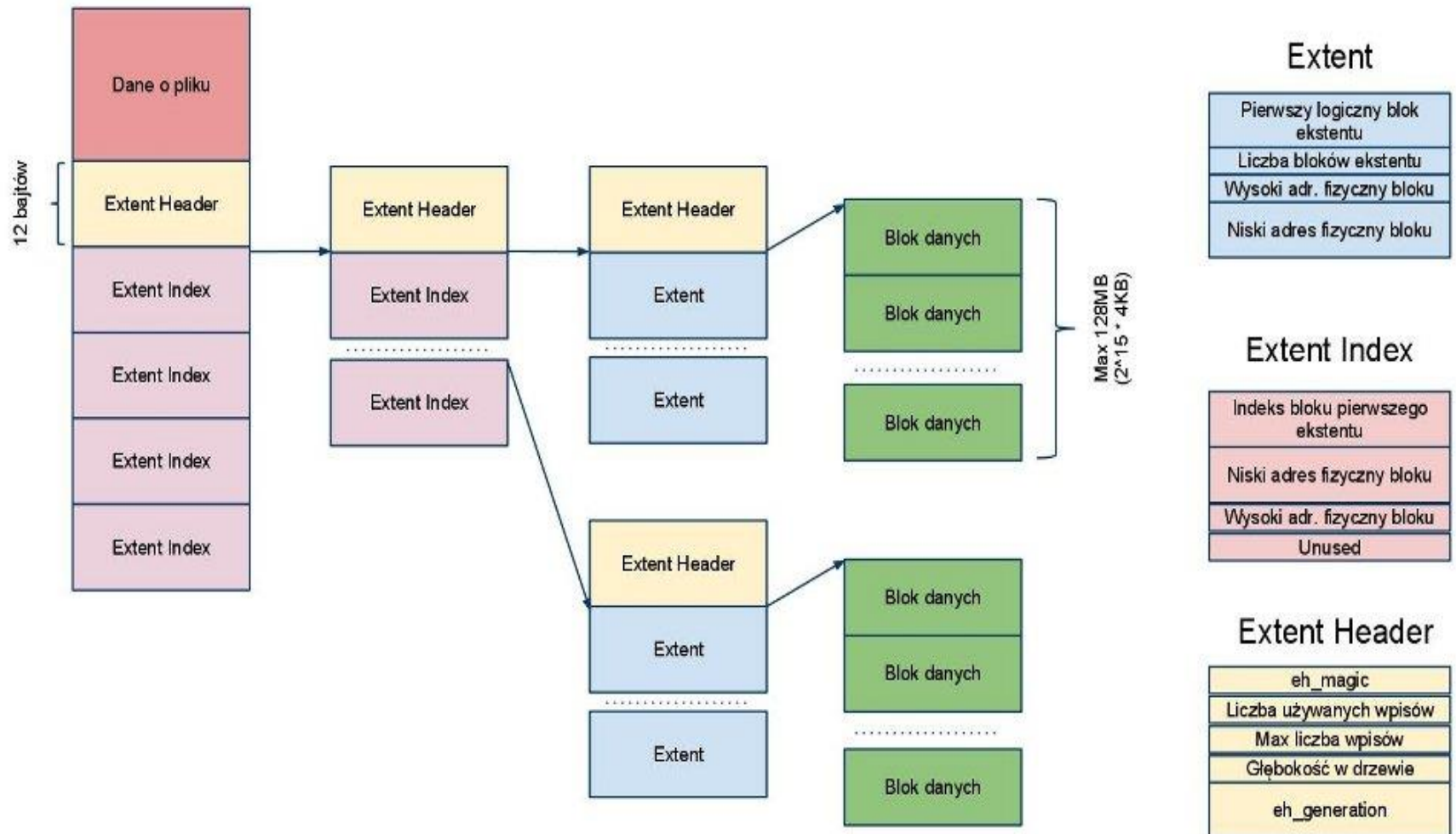
Przechowywanie plików o rozmiarze do 512 MB

```
struct ext4_extent_header {  
    __le16 eh_magic;  
    /* Magic number, probably will support different formats */  
    __le16 eh_entries;  
    /* Number of valid entries following the header */  
    __le16 eh_max;  
    /* Maximum number of entries that could follow the header */  
    __le16 eh_depth;  
    /* Depth of this extent node in the extent tree. */  
    __le32 eh_generation;  
    /* Generation of the tree */  
};
```

Przechowywanie plików o rozmiarze powyżej 512 MB

```
struct ext4_extent_idx {  
    __le32 ei_block;  
    /* index covers logical blocks from 'block' */  
    __le32 ei_leaf_lo;  
    /* pointer to the physical block of the next level. leaf  
    or next index could be there */  
    __le16 ei_leaf_hi;  
    /* high 16 bits of physical block */  
    __u16 ei_unused;  
};
```

Mapowanie bloków w ext4



Alokacja bloków

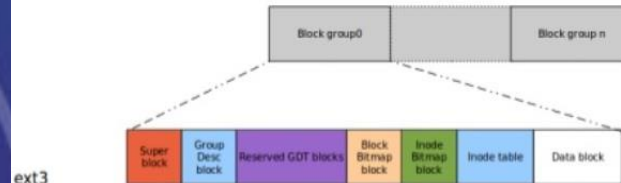
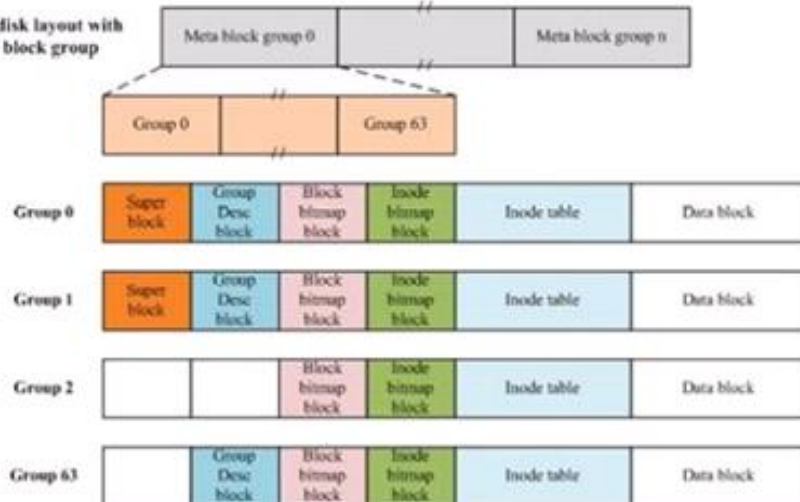
- Trwała prealokacja
- Jednoczesna alokacja wielu bloków (*multiblock allocation*)
- Opóźniona alokacja (*delayed allocation*)

Struktura danych na dysku

- Wyrównanie dla grupy 0 - 1024 bajty.
- Super-blok - 1 blok.
- Deskryptory grup - wiele bloków.
- Zarezerwowane bloki GDT - wiele bloków.
- Bitmapa bloków danych - 1 blok.
- Bitmapa i-węzłów - 1 blok.
- Tablica i-węzłów - wiele bloków.
- Tablica bloków danych - wiele bloków.

Alokacja bloków

B) ext4 disk layout with meta block group



ext3

ext4 flex_bg



Zamiana nazwy ścieżkowej na i - węzeł

Jądro chce odczytać plik np. /etc/hosts

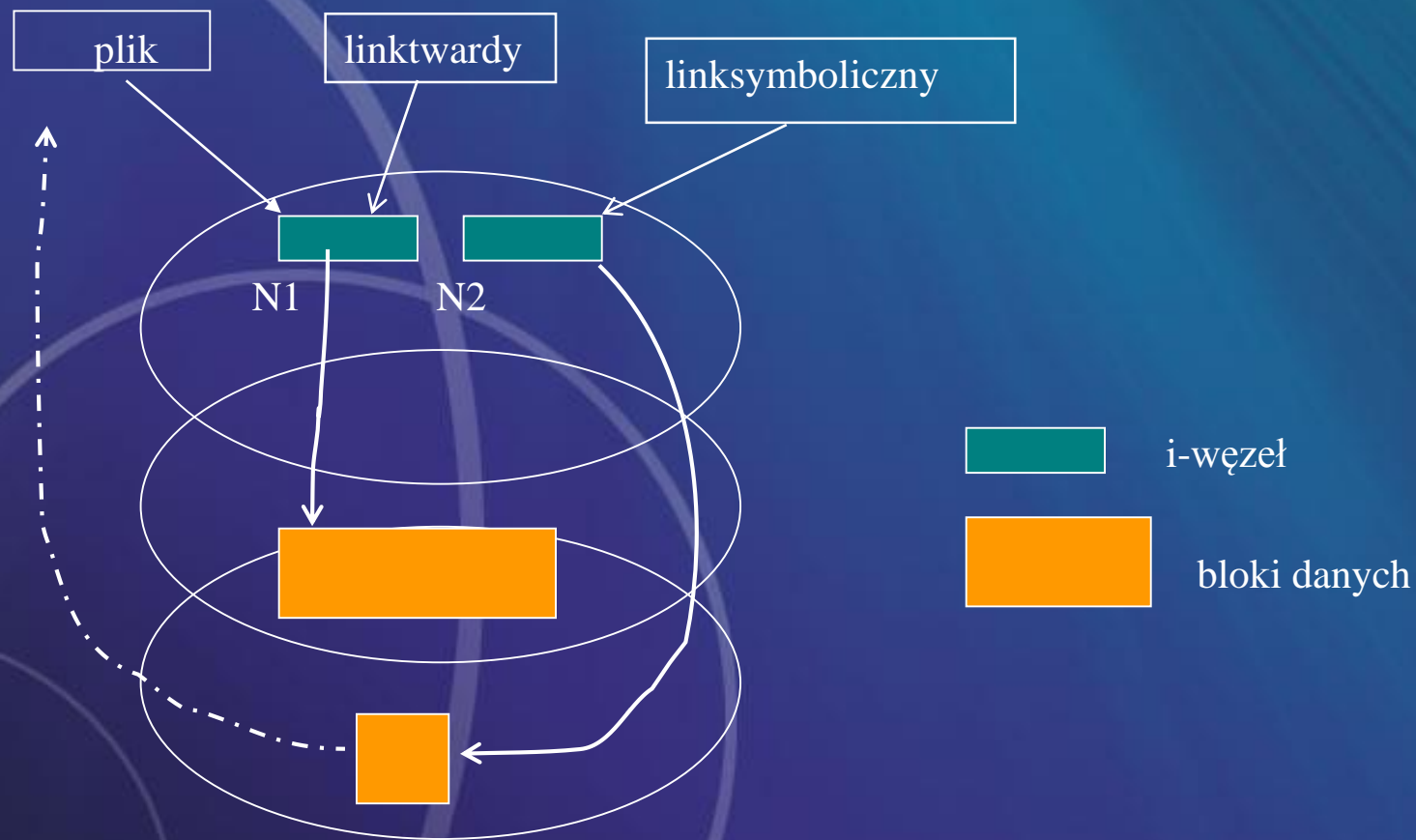
1. Odczytuje i-węzeł korzenia plików - nazwa zaczyna się od /
2. Sprawdza prawa dostępu oraz typ pliku
3. Ponieważ jest to katalog - wczytuje blok lub bloki z danymi
4. Szuka w nich nazwy etc.
5. Odczytuje ze struktury katalogu i-węzeł katalogu etc.
6. Odczytuje bloków dyskowe z zawartością katalogu etc.
7. Szuka ciągu znaków hosts.
8. Po odnalezieniu go pobiera i-węzeł.

Jądro kończy przekształcanie nazwy ścieżkowej w i-węzeł

Dowiązania do pliku

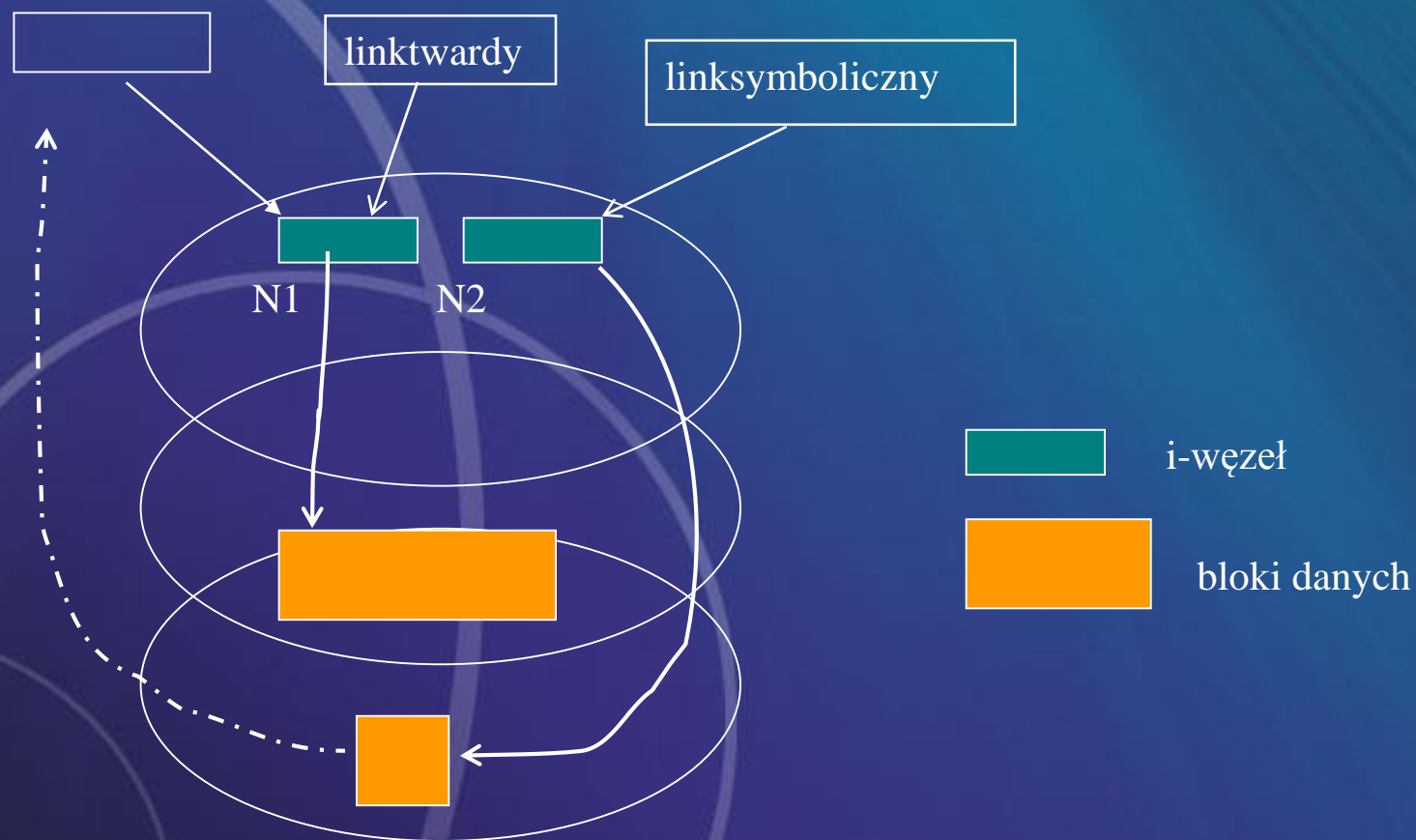
- Linki umożliwiają nadanie wielu nazw jednemu plikowi. Rozróżniamy dwa rodzaje linków:
 - linki twarde
 - linki symboliczne
- W różnych częściach systemu możemy utworzyć linki, które będą wskazywać na jeden plik. Nie musimy w ten sposób tworzyć wielu kopii tego samego pliku i możemy zaoszczędzić miejsce na dysku.
 - Link symboliczny jest plikiem, który wskazuje na nazwę innego pliku.
 - Link twardy umożliwia tworzenie kilku nazw dla jednego i-wężła.

Linki twarde i symboliczne

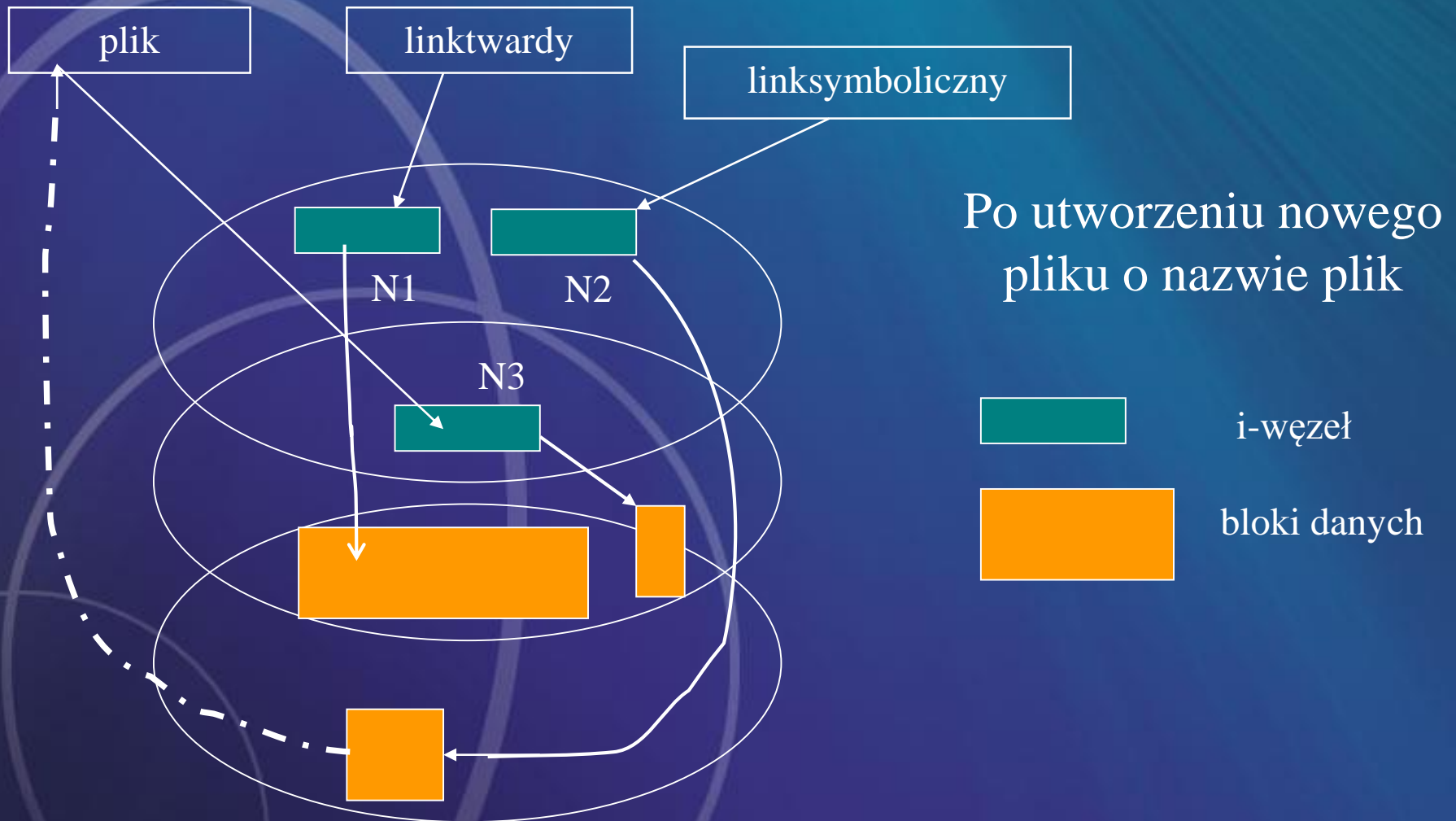


Linki twarde i symboliczne

Po usunięciu pliku plik



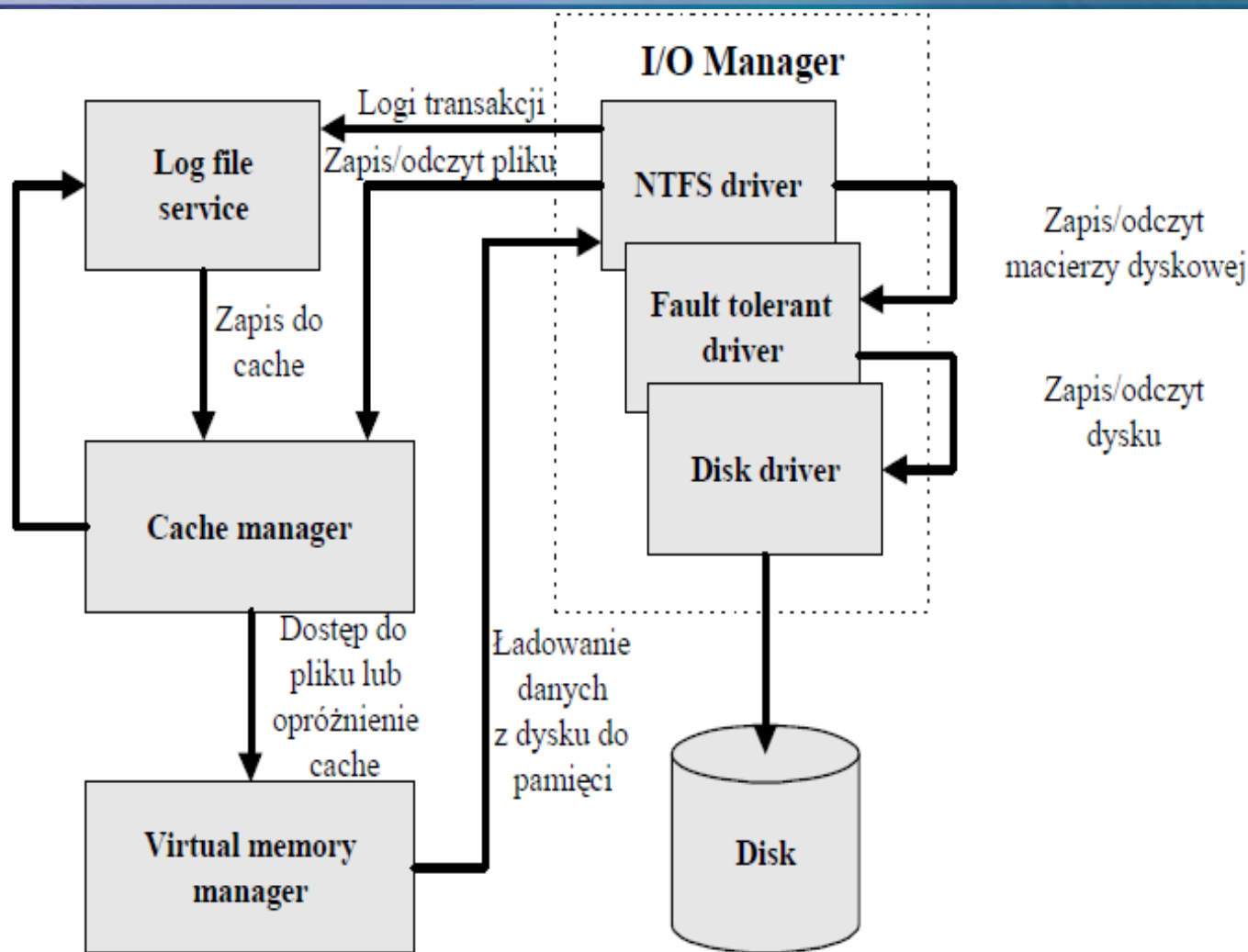
Dowiązania do pliku



NTFS – cechy ogólne

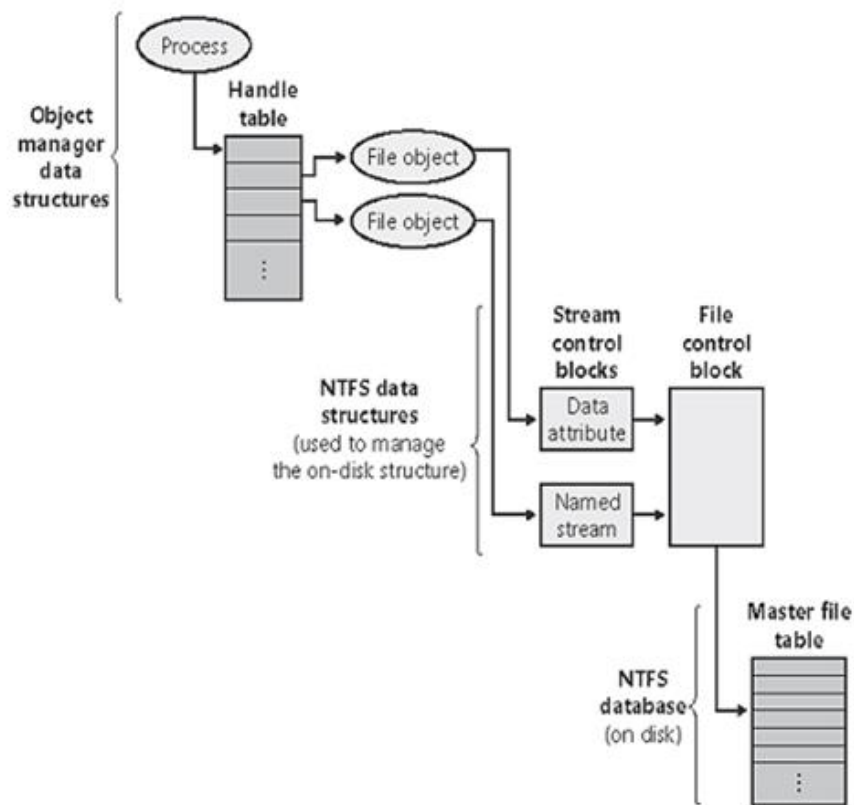
- Obiektowy systemu plików
- Bezpieczeństwo danych (Security)
- Nazwy
- Indeksowanie
- Quota
- Szyfrowanie
- Kompresja
- Pliki rzadkie
- Transakcyjne wykonywanie operacji We/Wy
- Przezroczysta dla użytkownika podmiana uszkodzonych sektorów
- Twarde dowiązania

Komponenty związane z NTFS



NTFS – struktura systemu

NTFS data structures



- Obiekt plikowy (file object), reprezentuje pojedyncze otwarcie pliku - wskazuje na blok SCB (Stream Control Block).
- Bloki SCB reprezentują indywidualne atrybuty plików - zawierają informacje jak znaleźć konkretny atrybut w pliku.
- Wszystkie bloki SCB danego pliku wskazują na wspólną strukturę FCB (File Control Block) – wskazuje na rekord w tablicy MFT (Master File Table).

Dyskowe struktury danych NTFS

- *Wolumen* - odpowiada logicznej partycji dysku i jest tworzony podczas formatowania dysku lub jego części dla NTFS.
- *Klaster* - logiczna jednostka danych na dysku, używana przez system plików - wielokrotność sektora równa potęgze liczby 2.
- Najistotniejszym elementem struktury wolumenu NTFS jest *tablica MFT (Master File Table)* - zaimplementowana jako tablica rekordów umieszczonych w pliku.

NTFS – struktura systemu

- Najmniejszą logiczną jednostką na dysku jest klaster.
- Wszelkie dane składowane są na dysku w postaci plików.
- Trzonem systemu NTFS jest **MFT**

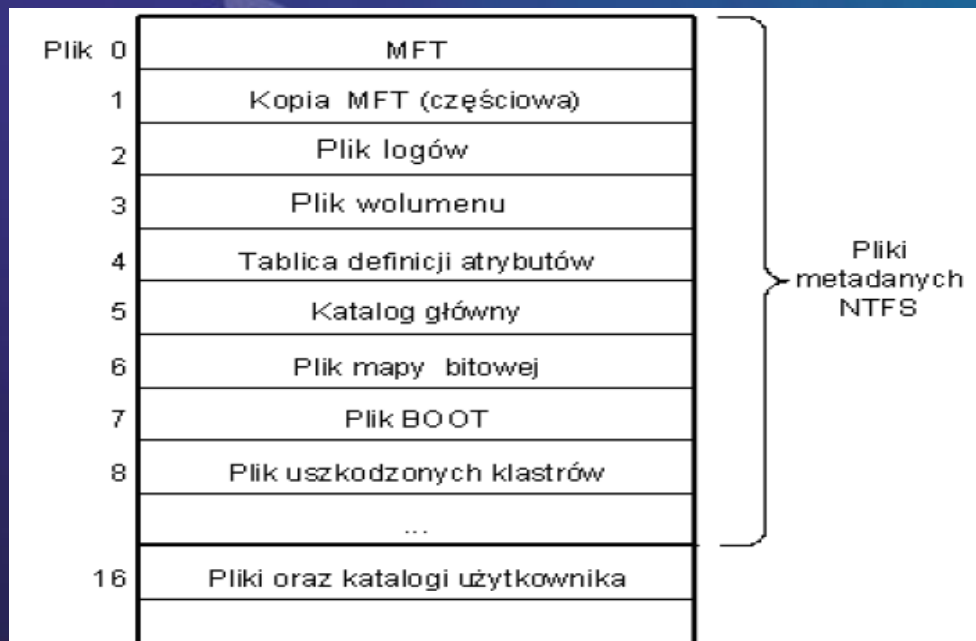
Struktura wolumenu:

BOOT SECTOR	MFT	SYSTEM FILES	DATA
-------------	-----	--------------	------

- Boot sector zawiera kod potrzebny do wystartowania systemu.
- MFT – tablica rekordów - pierwsze 16 wpisów w MFT jest zarezerwowane na metadane opisujące wolumen - nazwy tych plików zaczynają się od znaku \$.

Struktura tablicy MFT


- Rozmiar każdego rekordu jest stały i wynosi 1 kB.
- Logicznie tablica MFT zawiera jeden wiersz dla każdego pliku w wolumenie, włączając wiersz dla samej tablicy MFT.
- W każdym wolumenie NTFS znajduje się pliki metadanych (16 pozycji)
- Pozostałe pliki wolumenu NTFS są plikami użytkownika oraz katalogami.



NTFS – struktura systemu

Dla małych plików – ich
cała zawartość mieści
się w rekordzie w
tablicy MFT

Master File Table



Informacje standardowe	Nazwa pliku	Deskryptor bezpieczeństwa	Dane

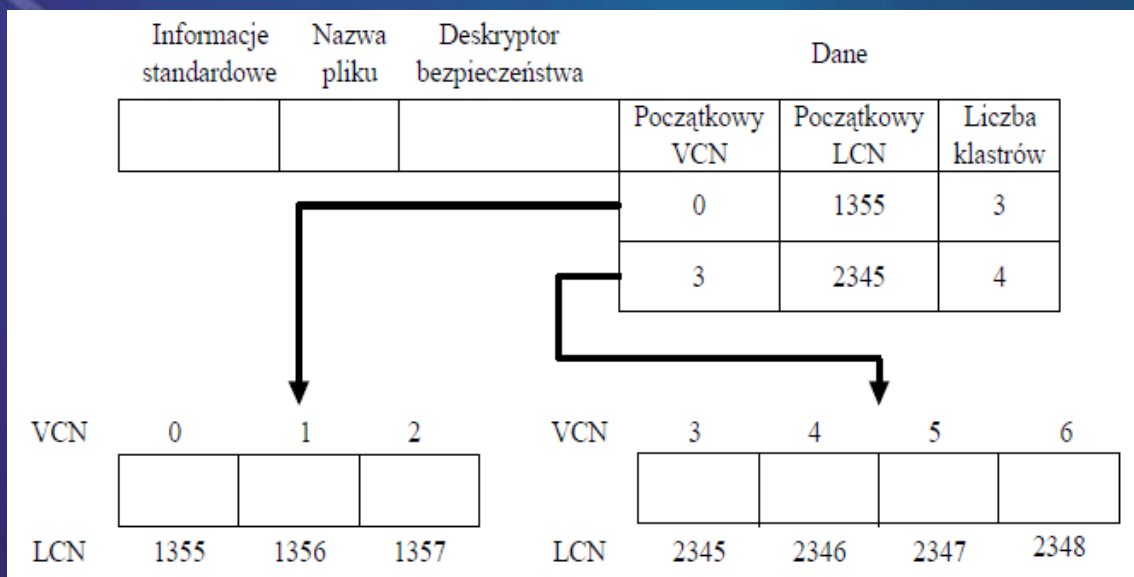
Analogicznie dla
niewielkiego
katalogu

MFT file record for a small directory

Standard information	Filename	Index root	
		Index of files	Empty
		file1, file2, file3, ...	

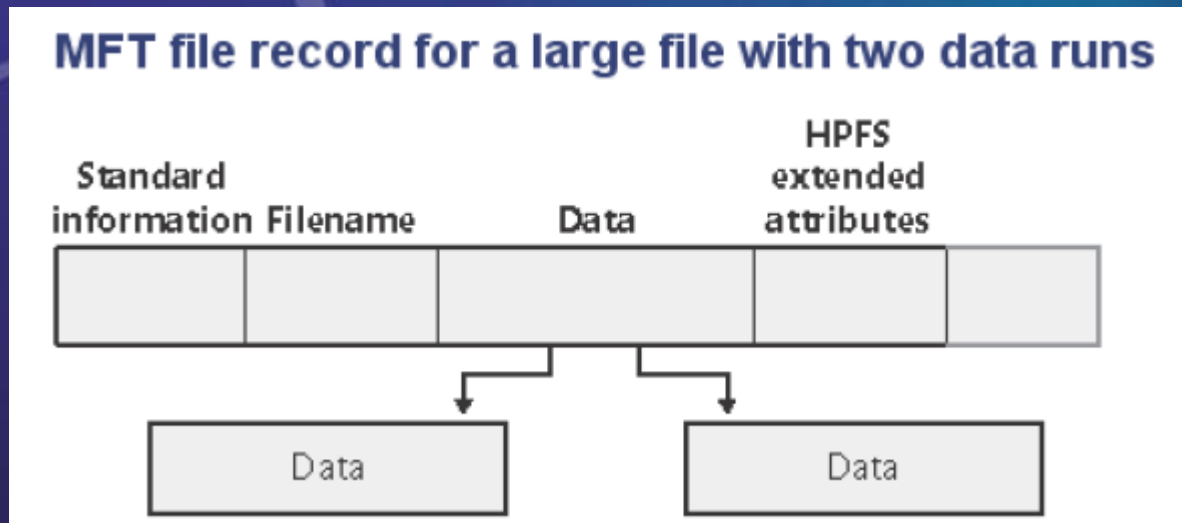
Standardowe atrybuty plików w NTFS

- Jeśli atrybut danych nie mieści się w rekordzie tablicy MFT, system NTFS alokuje klastry danego wolumenu poza tablicą MFT jako ekstenty.
- Nagłówek atrybutu, znajdujący się w rekordzie tablicy MFT, zawiera informację o rozmieszczeniu poszczególnych ekstentów w wolumenie.



Struktura plików NTFS

- Jeżeli plik jest na tyle duży że w rekordzie nie ma wystarczająco miejsca na pulę wskaźników, to w rekordzie jest wskaźnik do obszaru zawierającego pulę wskaźników.



Czym są dane pliku?

- NTFS definiuje pary: atrybut/wartość - zawartość pliku to jeden z atrybutów pliku o nazwie Data. Pozostałe atrybuty to :
 - nazwa pliku
 - deskryptor uprawnień
 - indeksy (tylko dla katalogów)
 - id obiektu
 - EFS – na potrzeby szyfrowania
 - informacje na temat wolumenu
 - standardowe informacje

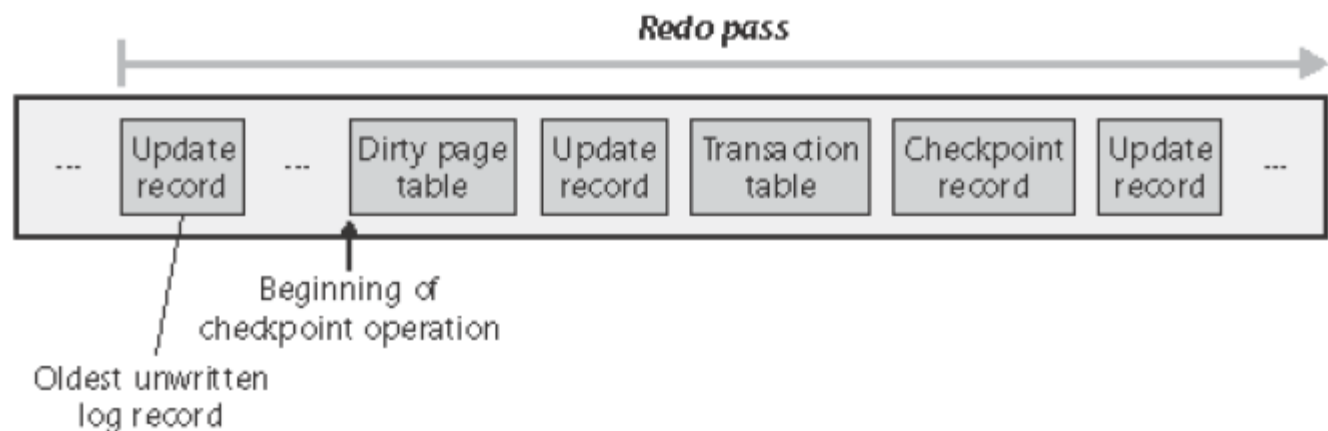
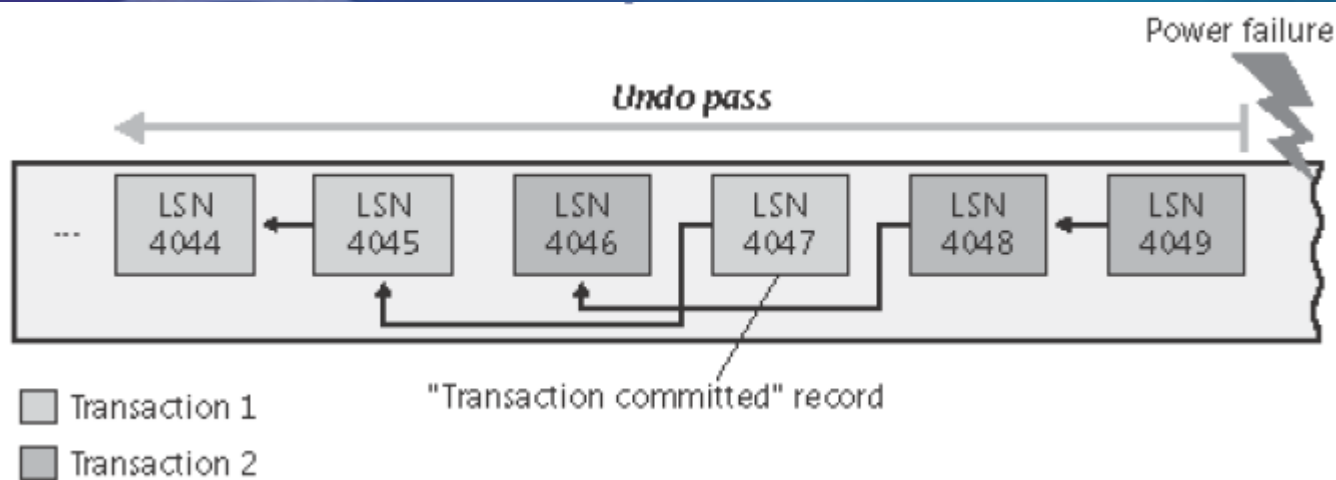
NTFS – odporność systemu

- System plików NTFS pozwala jedynie na pełną rekonstrukcję danych systemowych, czyli: katalogów, atrybutów bezpieczeństwa, mapy bitowej zajętych klastrów oraz pozostałych plików systemowych.

Odtwarzanie polega na:

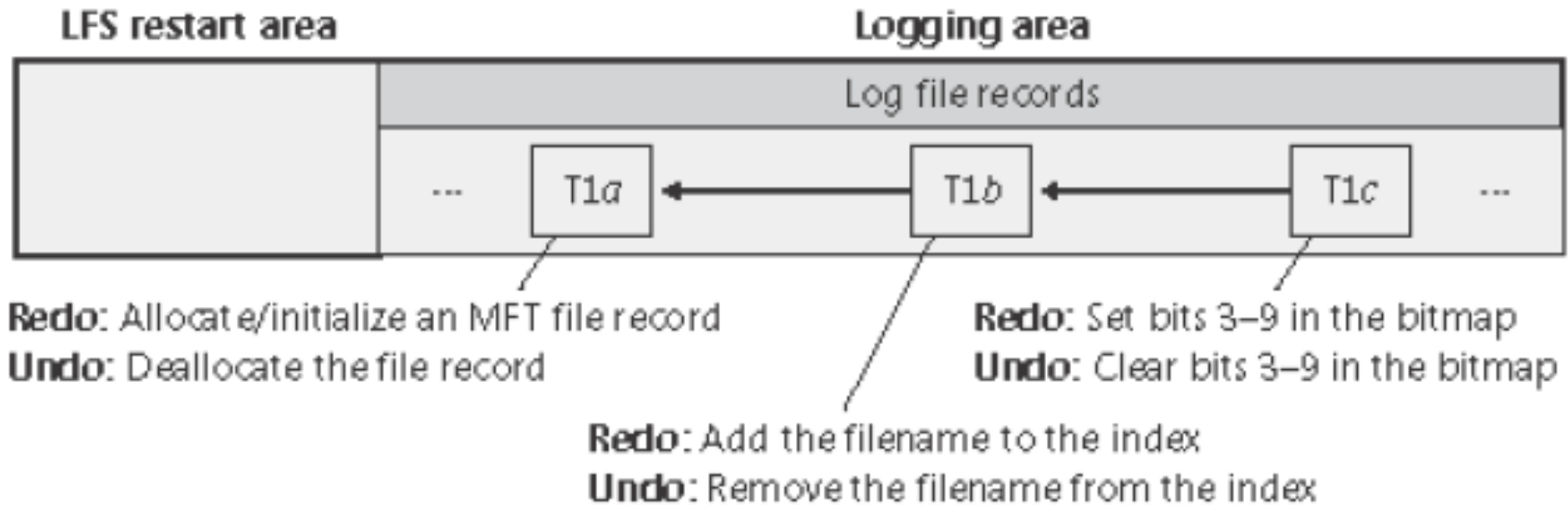
- Powtórzeniu wszystkich transakcji zatwierdzonych, zaczynając od najwcześniejszej.
- Unieważnieniu wszystkich transakcji niezakończonych, zaczynając od najpóźniejszej.

Odtwarzanie UNDO, REDO



Przebieg odtwarzania

Update records in the log file



NTFS – odporność systemu

- System odtwarzania bazuje w dużym stopniu na systemach odtwarzania spotykanych w systemach bazodanowych.
- W NTFS wprowadzono koncepcje transakcyjności operacji.
- Wszelkie operacje na danych wykonywane są w transakcjach.
- Zasadniczą techniką wspierającą odporność na awarie jest prowadzenie dziennika wszystkich operacji wykonanych na danych
- Dziennik zastosowany w NTFS jest typu undo/redo (unieważnienie/powtarzanie).

Odtwarzanie polega na:

- powtórzeniu wszystkich transakcji zatwierdzonych, zaczynając od najwcześniejszej
- unieważnieniu wszystkich transakcji niezakończonych, zaczynając od najpóźniejszej

Powłoka - shell

Najbardziej rozpowszechnionymi powłokami są :
bash oraz tcsh

- **bash** (**B**ourne-**A**gain **S**hell) został stworzony przez Briana Foxa i Cheta Rameya i udostępniony jest na zasadach licencji GNU. Jest w pełni kompatybilny z powłoką *sh* oraz zawiera wiele ciekawych rozwiązań przejętych od powłok Korn (*ksh* i *C csh*).
- **tcsh** - jest rozszerzoną wersją oryginalnego shella *csh* stworzonego przez Billy'ego Joy'a. Wiele konstrukcji z tego shella przypomina konstrukcje w języku C- stąd nazwa tej powłoki.

Zmienne powłoki

Najbardziej rozpowszechnionymi powłokami są :
bash oraz **tcsh**

- **bash** (Bourne-Again Shell) został stworzony przez Briana Foxa i Cheta Rameya i udostępniony jest na zasadach licencji GNU. Jest w pełni kompatybilny z powłoką *sh* oraz zawiera wiele ciekawych rozwiązań przejętych od powłok Korn (*ksh* i *C csh*).
- **tcsh** - jest rozszerzoną wersją oryginalnego shella *csh* stworzonego przez Billy'ego Joy'a. Wiele konstrukcji z tego shella przypomina konstrukcje w języku C- stąd nazwa tej powłoki.

Zmienne powłoki

- Lokalne
- Środowiskowe (sposób ich przekazywania jest analogiczny do przekazywania parametrów przez wartość – nowa powłoka dysponuje kopiami zmiennych środowiskowych)

nazwa zmiennej może się składać z dowolnych liter, cyfr oraz ze znaku "_", nie może jednak rozpoczynać się cyfrą

zmienne lokalne

zmienna=wartość

(obok operatora przypisania nie mogą znajdować się spacje!!)

- Należy się do nich odwoływać poprzez symbol **\$zmienna**

unset zmienna – usuwa zmienną

set wyświetla zmienne lokalne powłoki

Zmienne powłoki

Wartości przypisywane zmiennym mogą zawierać znaki, które mają szczególne znaczenie:

"...." – maskuje znaki specjalne (spacja, *, ?, [,], ., <, >, &, |)

'....' – działają jak "...", dodatkowo maskują znak \$

`.....` – wymuszają wykonanie polecenia

\ – maskuje jeden znak

export zmienna - tworzy zmienną środowiskową ze zdefiniowanej zmiennej

Linux definiuje specjalne zmienne powłoki, które są używane do konfigurowania powłoki:

env - wyświetla listę zdefiniowanych zmiennych specjalnych

Zmienne powłoki

Najważniejsze zmienne specjalne powłoki BASH:

HOME – ścieżka katalogu macierzystego,

USERNAME – nazwa użytkownika,

SHELL – ścieżka powłoki logowania,

PATH – lista katalogów rozdzielonych dwukropkami, które są przeszukiwane w celu znalezienia polecenia do uruchomienia,

CDPATH – lista katalogów przeszukiwanych podczas wykonywania komendy cd,

HISTSIZE – ilość zdarzeń historii,

HISTFILE – nazwa pliku historii zdarzeń,

HISTFILESIZE – rozmiar pliku historii,

Zmienne powłoki

- BASH używa kilku dodatkowych zmiennych określających właściwości powłoki, będącymi opcjami, które mogą być włączone lub wyłączone. Ustawienie zmiennej dokonuje się poleceniem:

set +o zmienna

set -o zmienna

- Znaczenie poszczególnych zmiennych:
 - **ignoreeof** – blokuje możliwość wylogowania się za pomocą [CTRL+d] (domyślnie do 10 razy). Można nadać jej dowolną wartość (np. ignoreeof=30),
 - **noclobber** – zapobiega nadpisaniu plików podczas przekierowania (>),
 - **noglob** – blokuje rozwijanie znaków specjalnych.

Historia

history - wyświetla listę ponumerowanych poleceń (zdarzeń) - można się do nich odwoływać używając poleceń z wykrzyknikiem.

!3 – wyświetla i wykonuje polecenie historii nr 3

!ab – wyświetla i wykonuje ostatnio wydane polecenie rozpoczynające się ciągiem znaków ab

!?ab? – wyświetla i wykonuje ostatnio wydane polecenie zawierające ciąg znaków ab

!-3 – wyświetla i wykonuje trzecie licząc od tyłu wydane polecenie

Polecenia historii można składać. Jeśli np. polecenie o numerze 100 ma postać **ls**, wówczas **!100 *.c** jest równoważne poleceniu **ls *.c**

Możliwe jest także odwołanie do parametrów ostatnio wydanego polecenia poprzez **!!*** lub **!***

Aliasy

- Aliasy, czyli utożsamienia, to dodatkowe nazwy dla najczęściej wykonywanych poleceń.
- Polecenie **alias**:
 - wyświetla listę zdefiniowanych aliasów
 - definiuje nowe aliasy

Przykłady

alias kto=who

alias l='ls -li' - l

jeśli w poleceniu występują spacje, należy użyć pojedynczych cudzysłowów,

alias lc='ls -l *.c'

alias cp='cp -i' nazwa polecenia również może być aliasem.

- Aby usunąć zdefiniowany alias, należy użyć polecenia **unalias**

Pliki startowe powłoki bash

- **.bash_profile** - plik inicjacyjny logowania powłoki BASH, wykonywany automatycznie w czasie logowania użytkownika,
- **.bashrc** - wykonywany jest przy każdym uruchomieniu nowego shella powłoki,
- Oba te pliki znajdują się w katalogu domowym użytkownika.
- **bash_logout** - wykonywany w czasie wylogowywania się użytkownika.