



# Linux cz.2: terminal, użytkownicy, grupy

Wykład: terminal, konsola, GUI, pseudoterminal, proces, zadanie, usługa, daemon, użytkownicy w Linuxie, grupy, logowanie, uwierzytelnianie, autoryzacja, moduł ładowalny, PAM

# Wirtualny terminal i konsola

## Terminal

dawniej, przez terminal komputerowy rozumieliśmy urządzenie, które pozwalało użytkownikowi na zdalne wykorzystanie mocy obliczeniowej komputera, do którego połączony był terminal.

Dziś pod pojęciem terminal rozumiemy emulację terminala sprzętowego, czyli wirtualny terminal udostępniany przez system operacyjny, służący użytkownikowi do porozumiewania się z komputerem w trybie tekstowym

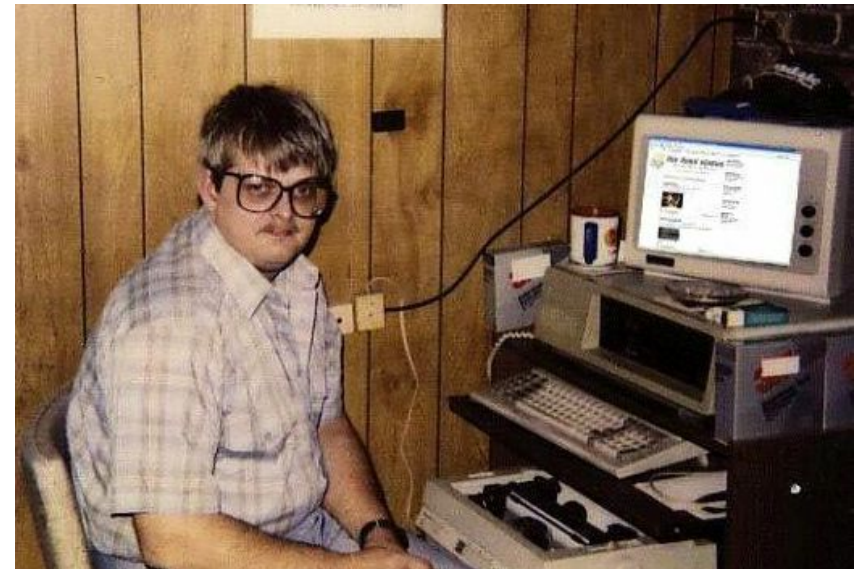


# Wirtualny terminal i konsola

## Konsola

tak określano dawniej terminal, na którym pracował administrator systemu. Dziś konsola podobnie jak terminale również jest emulowana (czyli istnieje tylko wirtualnie) w systemie. Każdy terminal zostaje konsolą po zalogowaniu się do niego roota

Ciekawostka: do dziś istnieje konwencja, która terminem konsoli pozwala nazywać tylko ten terminal, na którym zalogowany jest administrator (root). Nie wszyscy użytkownicy znają jednak etymologię tych słów i dlatego używają ich zamiennie



# Terminal wirtualny (1)

Podłączanie terminali sprzętowych jest nadal możliwe, Linux bowiem obsługuje takie urządzenia. W systemie Linux terminal oznaczany jest skrótem **tty** pochodzącym od słowa **teletypewriter**, oznaczającego dalekopis.

Terminale wirtualne działają we współpracy z powłoką systemową, która pobiera wpisane przez użytkownika w terminalu dane, przetwarza je i wypisuje w terminalu wyniki działania programów lub wbudowanych poleceń.

W systemach Linux wystarczy wpisać w terminalu wirtualnym (po zalogowaniu się) nazwę zainstalowanej powłoki aby rozpocząć w niej pracę.

Zmiany powłoki można dokonać niemal w każdym momencie pracy w terminalu wirtualnym

# Terminal wirtualny (2)

W systemach uniksowych terminal wirtualny traktowany jest jak fizyczne urządzenie wejścia-wyjścia. Posiada bowiem swój plik specjalny w katalogu `/dev`

Linux udostępnia użytkownikowi kilka terminali wirtualnych, pomiędzy którymi może się przełączać używając odpowiednich kombinacji klawiszy (o czym szerzej na następnym slajdzie).

W katalogu  `dev` znajdują się pliki `tty1`, `tty2`, `tty3`, itd. Odpowiadają one kolejnym terminalom wirtualnym.

Plik `tty` odpowiada bieżącemu terminalowi, na którym w danej chwili pracuje użytkownik

# Terminal wirtualny (3)

Jeśli użytkownik aktualnie pracuje w środowisku graficznym, to na terminal wirtualny o danym numerze przełączy go kombinacja klawiszy **[Ctrl]+[Alt]+[F<numer konsoli>]**

Na przykład kombinacja **[Ctrl]+[Alt]+[F1]** przełącza na pierwszy terminal wirtualny

Jeśli użytkownik pracuje aktualnie w środowisku tekstowym, to na terminal wirtualny o danym numerze przełączy go kombinacja klawiszy **[Alt]+[F<numer konsoli>]**

# Terminal wirtualny (4)

Aby rozpocząć pracę na danej konsoli wirtualnej należy się na niej zalogować.

Na każdym terminalu wirtualnym może zalogować się inny użytkownik.

Ten sam użytkownik może być zalogowany na kilku terminalach równocześnie.

Wszystkie utworzone w ten sposób sesje są równocześnie obsługiwane przez system operacyjny.

# Graphical User Interface

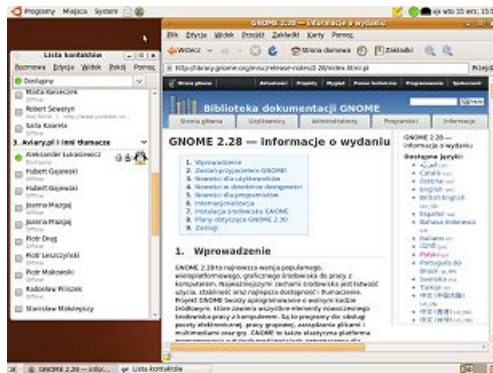
## Graficzny interfejs użytkownika (GUI)

często nazywany też środowiskiem graficznym – określa sposób prezentowania informacji przez komputer oraz interakcji z użytkownikiem

System operacyjny pozbawiony środowiska graficznego to **system tekstowy**.

Systemy komunikujące się z użyciem środowiska graficznego to **systemy graficzne**.

Przykładowe środowiska GUI:



GNOME - Linux/Unix



KDE - Linux/Unix



Aero - Windows Vista/7



# Pseudoterminal

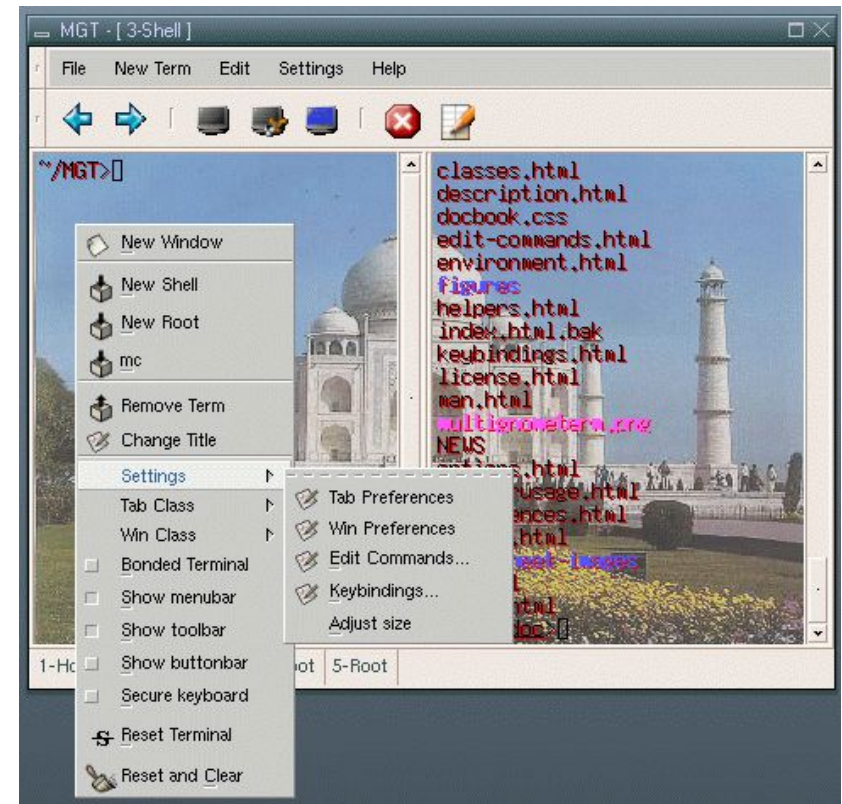
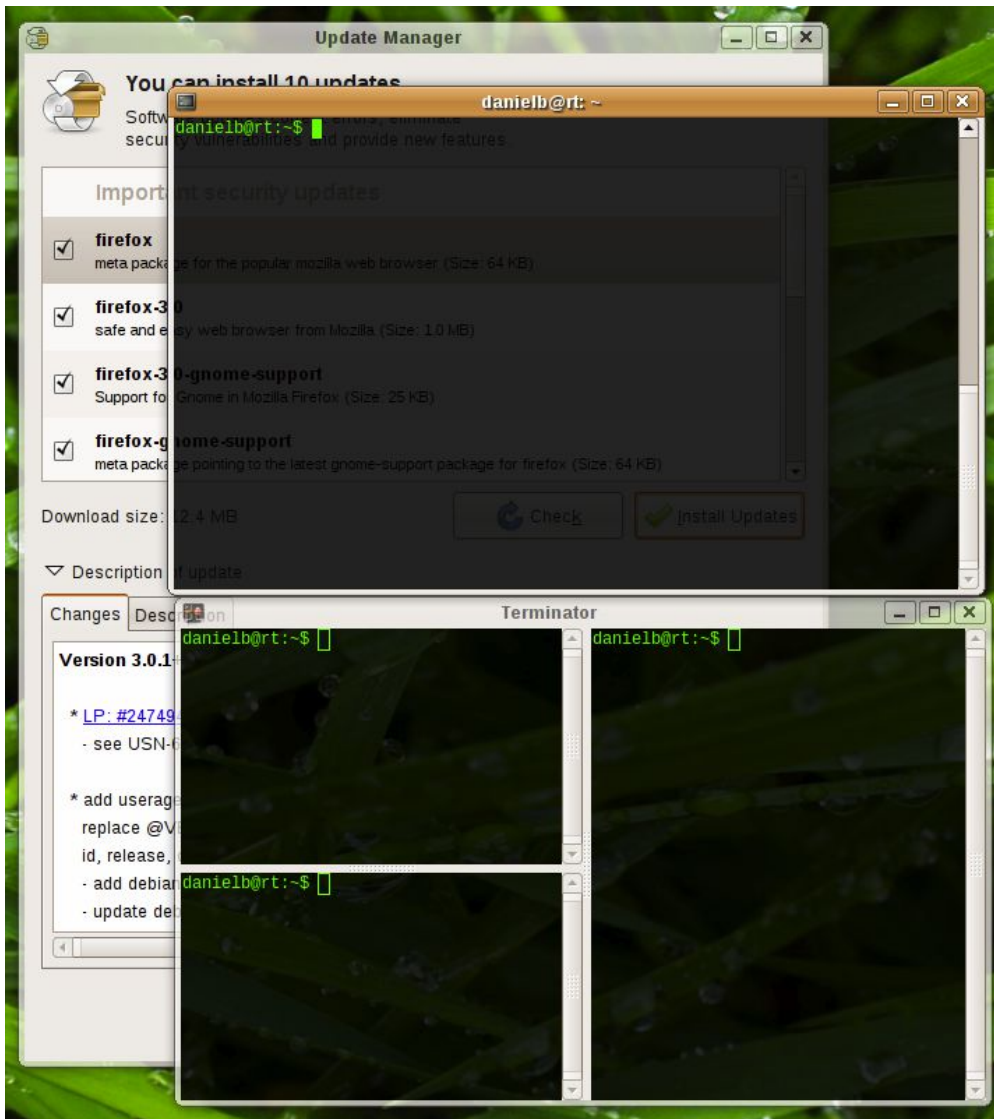
Korzystanie z terminali wirtualnych było popularne w czasach, gdy jeszcze nie stworzono środowisk graficznych.

Dzisiaj, częściej niż z terminali wirtualnych korzysta się z emulatorów wirtualnych terminali uruchamianych w oknach programów środowiska graficznego (nazywane są wówczas pseudoterminalami)

## Pseudoterminal

Programowa emulacja terminala wirtualnego w środowisku graficznym

# Graficzne emulatory terminali



# Graficzne emulatory terminali

Istnieje kilka programów udostępniających emulację terminali wirtualnych w środowisku graficznym, oto niektóre z nich:

- **xterm** - standardowy emulator terminala dla XWindow,
- **gnome-terminal** - emulator terminala dla środowiska graficznego GNOME
- **konsole** - emulator terminala środowiska graficznego KDE

# Pamiętacie, czy już zapomnieliście?

- Linux jest systemem wielodostępnym i wielozadaniowym.
- Jako system **wielozadaniowy**, pozwala na wykonywanie kilku zadań (procesów) w tym samym czasie - gdy jedno zadanie jest wykonywane, można pracować nad następnym
- Jako system **wielodostępny**, pozwala na logowanie się do systemu kilku użytkowników w tym samym czasie, a każdy użytkownik współdziela z systemem poprzez swój własny terminal



# Procesy w systemie

## Proces

egzemplarz danego programu posiadający własną przestrzeń adresową

- Każdy nowo powstały proces otrzymuje unikalny numer, który go jednoznacznie identyfikuje (PID)
- W celu wykonania programu system operacyjny przydziela procesowi zasoby (pamięć, czas procesora)

# Działania na procesach

Proces można:

- widzieć na ekranie monitora (efekty pracy procesu będą widoczne). Taki proces nazywamy **pierwszoplanowym**
- przenieść **w tło** (ang. background) przy użyciu **Ctrl+z** oraz komendy **bg** dla danych wyjścia - proces będzie pracował, ale bez pokazywania efektów na ekranie). Taki proces nazywamy **drugoplanowym**
- zawiesić, czyli wstrzymać (ang. suspend). Taki proces również nazywamy **drugoplanowym**
- zabić, czyli zakończyć (ang. kill)

# Zadania. Usługi. Demony

## Zadanie

to proces drugoplanowy, który może być "zawieszony" lub działający "w tle".

## Usługa

czyli serwis systemowy - w Linuksie są to programy uruchamiane podczas startu systemu w celu wykonania pewnych czynności

## Demon

usługa działająca przez cały czas pracy systemu (działa jako proces w tle, czyli nie jest podłączona do żadnego terminala)

# Użytkownicy w Linuxie (1)

superużytkownik = administrator = root

użytkownik mający **wszystkie** prawa w systemie (do modyfikowania lub usuwania plików, uruchamiania programów i wykonywania dowolnych poleceń możliwych do wykonania w systemie)

Konto administratora tworzone jest podczas instalacji systemu. Tworzenie innych kont nie jest wymagane, ale mocno zalecane.

**Dla dobra systemu nie powinno się przesiadywać cały czas na koncie roota**

Zatem nawet jeśli do Twojego komputera dostęp masz tylko Ty to i tak powinieneś utworzyć dla siebie konto zwykłego użytkownika





# Użytkownicy w Linuxie (2)

## użytkownik systemowy

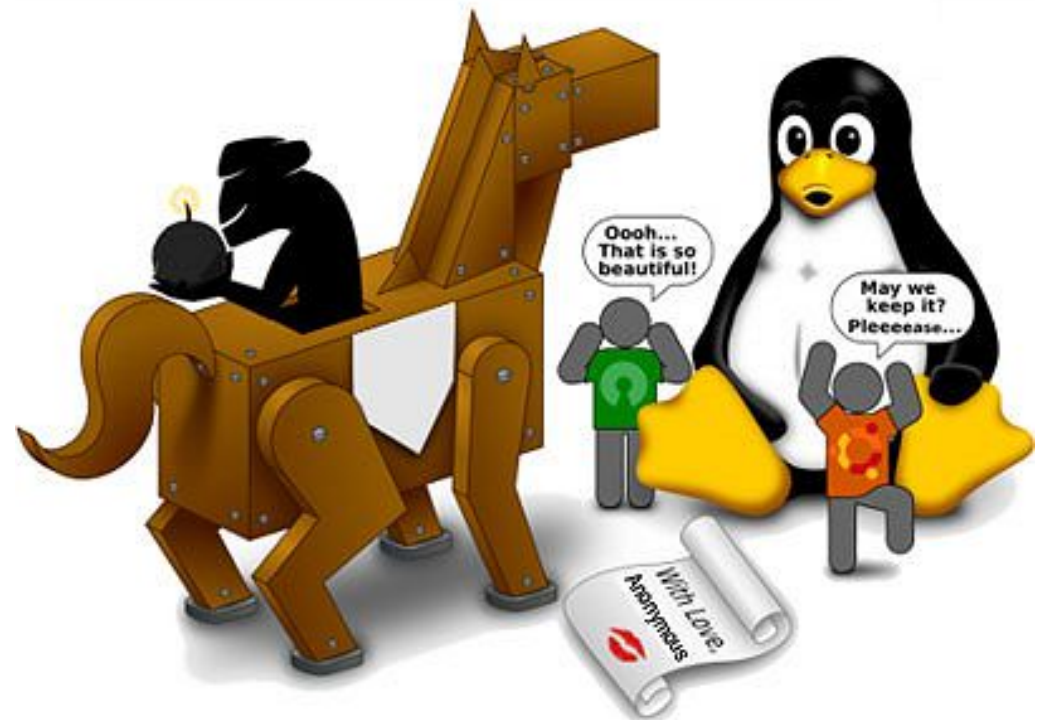
to każdy proces w systemie, któremu system przyznał odpowiednie pliki, katalogi oraz przestrzeń adresową, które są mu niezbędne do pracy. Użytkownikiem systemowym nie może być osoba fizyczna



# Użytkownicy w Linuxie (3)

## użytkownik zwykły

jest to osoba fizyczna, która wykorzystuje system do realizacji zadań **niezwiązanych z administracją systemu**



# Zarządzanie użytkownikami

- Każdy użytkownik otrzymuje unikalny numer, który go jednoznacznie identyfikuje w systemie – UID (**U**ser **I**D)
- Każdy użytkownik mający konto w systemie otrzymuje m.in. własny **katalog domowy**, stanowiący jego środowisko pracy
- Stworzenie nowego użytkownika polega na dodaniu go do bazy użytkowników i następnie utworzeniu dla niego katalogu domowego
- Do zarządzania użytkownikami są używane polecenia, np: **useradd**, **passwd**, **usermod**, **userdel**, **su**

# Plik haseł (baza użytkowników)

Podstawową bazą użytkowników w Linuxie jest tzw. plik haseł: `/etc/passwd`. Każdy wiersz tego pliku zawiera informacje o jednym koncie i składa się z 7 wartości oddzielonych dwukropkami:

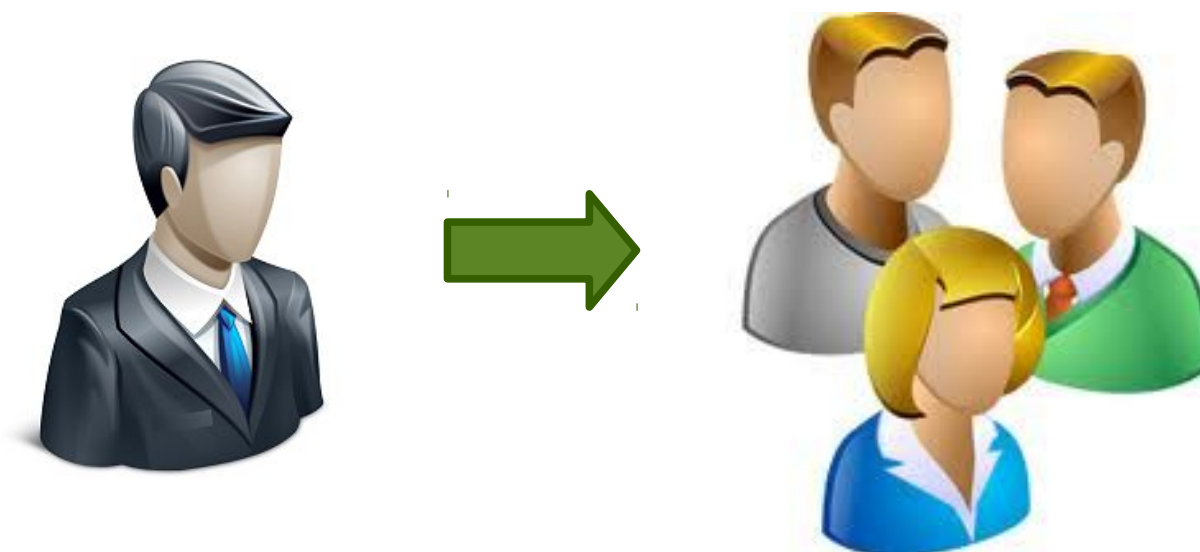
1. Nazwa użytkownika
2. `x` lub `*` oznaczający, że hasło znajduje się w innym pliku
3. UID
4. GID
5. Imię i nazwisko oraz opis użytkownika
6. Ścieżka katalogu domowego
7. Powłoka logowania (program uruchamiany po zalogowaniu)

A gdzie te hasła? Znajdują się w postaci zakodowanej w pliku `/etc/shadow`, który może być odczytany tylko przez roota

# Grupy użytkowników

## Grupy

tworzone są dla użytkowników charakteryzujących się daną cechą. Dzięki grupom możliwe jest ustalenie uprawnień do zasobów dla większego grona użytkowników jednocześnie - nie zaś dla każdego indywidualnie



# Zarządzanie grupami

- Każda grupa otrzymuje unikalny numer, który jednoznacznie identyfikuje ją w systemie – GID (Group ID)
- Jeżeli do istniejącej grupy zostanie dołączony kolejny użytkownik, będzie on miał takie same prawa jak pozostali
- Każdy z użytkowników musi należeć przynajmniej do jednej grupy
- Do zarządzania grupami są używane polecenia, np: `addgroup`, `delgroup`, `chgrp`

# Plik informacji o grupach

Wszystkie grupy zapisane są w pliku: `/etc/group`.

Każdy wiersz tego pliku zawiera informacje o jednej grupie i składa się z 4 wartości oddzielonych dwukropkami:

1. Nazwa grupy
2. Hasło - jeżeli wartość jest pusta, to hasło nie jest wymagane
3. GID
4. Lista użytkowników należących do tej grupy

# Logowanie

## Logowanie

proces **uwierzytelniania** i **autoryzacji** użytkownika komputera, polegający najczęściej na podaniu identyfikatora użytkownika oraz hasła uwierzytelniającego w celu uzyskania dostępu oraz określonych uprawnień do korzystania z systemu komputerowego





# Uwierzytelnianie. Autoryzacja

## Uwierzytelnianie

proces polegający na zweryfikowaniu wiarygodności użytkownika próbującego uzyskać dostęp do zasobów systemu

## Autoryzacja (upoważnienie)

proces pozwalający zdecydować do jakich zasobów dany użytkownik może uzyskać dostęp i jakie operacje może wykonać w systemie

# Linux PAM

## System PAM

(ang. **P**luggable **A**uthentication **M**odules) biblioteki pozwalające na uwierzytelnienie użytkownika przez programy w systemie

PAM zapewnia różne rodzaje autoryzacji, poczynając od trywialnej autoryzacji poprzez hasło, a kończąc na autoryzacji poprzez głos, linie papilarne lub jednorazowe hasło

Dodatkowo PAM może zadbać automatycznie o zarządzanie hasłami i ich automatyczną aktualizację, blokowanie kont po upływie określonego czasu bezczynności, obsługę zablokowanych kont, logowanie roota tylko do jednego terminalu, zapisywanie do logów danych wymienionych z programem po zalogowaniu



# Moduł ładowalny

Jądro systemu Linux potrafi ładować i usuwać dowolną część swojego kodu stosownie do potrzeb. Osobno ładowane części kernela noszą nazwę **modułów ładowalnych**

Linux jest kompilowany i łączony (linkowany) dynamicznie. Oznacza to, że w każdej chwili możemy załadować do niego jakiś fragment kodu, bądź go usunąć. Pozwala to na między innymi zmniejszenie rozmiarów kernela, skracając tym samym czas ładowania go do pamięci

Aby przekonać się, jakie moduły mamy obecnie do dyspozycji, wystarczy wydać polecenie **lsmod**

Można powiedzieć, że moduły odpowiadają sterownikom w terminologii DOS/Windows, jednak mogą one obsługiwać nie tylko urządzenia



# Źródła

Systemy operacyjne i sieci komputerowe. Cz. 1.  
WSIP. Krzysztof Pytel, Sylwia Osetek

Systemy operacyjne i sieci komputerowe. Cz. 2.  
WSIP. Krzysztof Pytel, Sylwia Osetek