

# Dystrybucje Linuxa

M.Barczyński W.Bederski M.Startek

Systemy Operacyjne 2005/2006

# Plan prezentacji

- 1 Wstęp
  - Historia GNU/Linux
  - Słów kilka o dystrybucjach
- 2 Porównanie dystrybucji
  - Osiołkowi w żłoby dano, czyli co dla kogo.
  - Systemy zarządzania pakietami
- 3 Standardy w systemach \*nix'owych
  - POSIX
  - SUS
  - LSB
  - FHS
  - Zgodność dystrybucji
- 4 Absolutne minimum

## Co to jest dystrybucja?

Jak podaje Wikipedia: "dystrybucja (...) systemu operacyjnego Linux to zestaw programów rozpowszechnianych łącznie i dający po zainstalowaniu gotowy do użycia system".

# Plan prezentacji

- 1 Wstęp
  - Historia GNU/Linux
  - Słów kilka o dystrybucjach
- 2 Porównanie dystrybucji
  - Osiołkowi w żłoby dano, czyli co dla kogo.
  - Systemy zarządzania pakietami
- 3 Standardy w systemach \*nix'owych
  - POSIX
  - SUS
  - LSB
  - FHS
  - Zgodność dystrybucji
- 4 Absolutne minimum

# GNU

## GNU's Not UNIX

- Richard Stallman 27 września 1983 roku ogłasza manifest GNU
- ...ale prace zaczynają się dopiero 5 stycznia 1984...
- ...celem jest stworzenie otwartego systemu operacyjnego.
- Do 1990 GNU miało już: Emacs'a, gcc, T<sub>E</sub>X, XWindowSystem<sup>1</sup>, i praktycznie wszystkie niezbędne biblioteki...
- ...brakowało tylko skończonego Trix'a, Mach'a, Hurd'a itd.
- wtedy pojawia się...

---

<sup>1</sup>T<sub>E</sub>Xi X'y nie były napisane przez grupę GNU, lecz powstały wcześniej

# GNU

## GNU's Not UNIX

- Richard Stallman 27 września 1983 roku ogłasza manifest GNU
- ...ale prace zaczynają się dopiero 5 stycznia 1984...
- ...celem jest stworzenie otwartego systemu operacyjnego.
- Do 1990 GNU miało już: Emacs'a, gcc, T<sub>E</sub>X, XWindowSystem<sup>1</sup>, i praktycznie wszystkie niezbędne biblioteki...
- ...brakowało tylko skończonego Trix'a, Mach'a, Hurd'a itd.
- wtedy pojawia się...

---

<sup>1</sup>T<sub>E</sub>Xi X'y nie były napisane przez grupę GNU, lecz powstały wcześniej

# GNU

## GNU's Not UNIX

- Richard Stallman 27 września 1983 roku ogłasza manifest GNU
- ...ale prace zaczynają się dopiero 5 stycznia 1984...
- ...celem jest stworzenie otwartego systemu operacyjnego.
- Do 1990 GNU miało już: Emacs'a, gcc, T<sub>E</sub>X, XWindowSystem<sup>1</sup>, i praktycznie wszystkie niezbędne biblioteki...
- ...brakowało tylko skończonego Trix'a, Mach'a, Hurd'a itd.
- wtedy pojawia się...

---

<sup>1</sup>T<sub>E</sub>Xi X'y nie były napisane przez grupę GNU, lecz powstały wcześniej

# GNU

## GNU's Not UNIX

- Richard Stallman 27 września 1983 roku ogłasza manifest GNU
- ...ale prace zaczynają się dopiero 5 stycznia 1984...
- ...celem jest stworzenie otwartego systemu operacyjnego.
- Do 1990 GNU miało już: Emacs'a, gcc, T<sub>E</sub>X, XWindowSystem<sup>1</sup>, i praktycznie wszystkie niezbędne biblioteki...
- ...brakowało tylko skończonego Trix'a, Mach'a, Hurd'a itd.
- wtedy pojawia się...

---

<sup>1</sup>T<sub>E</sub>Xi X'y nie były napisane przez grupę GNU, lecz powstały wcześniej



# GNU

## GNU's Not UNIX

- Richard Stallman 27 września 1983 roku ogłasza manifest GNU
- ...ale prace zaczynają się dopiero 5 stycznia 1984...
- ...celem jest stworzenie otwartego systemu operacyjnego.
- Do 1990 GNU miało już: Emacs'a, gcc, T<sub>E</sub>X, XWindowSystem<sup>1</sup>, i praktycznie wszystkie niezbędne biblioteki...
- ...brakowało tylko skończonego Trix'a, Mach'a, Hurd'a itd.
- wtedy pojawia się...

---

<sup>1</sup>T<sub>E</sub>Xi X'y nie były napisane przez grupę GNU, lecz powstały wcześniej

# GNU

## GNU's Not UNIX

- Richard Stallman 27 września 1983 roku ogłasza manifest GNU
- ...ale prace zaczynają się dopiero 5 stycznia 1984...
- ...celem jest stworzenie otwartego systemu operacyjnego.
- Do 1990 GNU miało już: Emacs'a, gcc, T<sub>E</sub>X, XWindowSystem<sup>1</sup>, i praktycznie wszystkie niezbędne biblioteki...
- ...brakowało tylko skończonego Trix'a, Mach'a, Hurd'a itd.
- wtedy pojawia się...

---

<sup>1</sup>T<sub>E</sub>Xi X'y nie były napisane przez grupę GNU, lecz powstały wcześniej

# Linux

- 17 września 1991 Linus Torvalds zamieszcza pierwszą wersję Linuxa w sieci.
- Hobby: poznanie architektury PC i języka C
- Terminal tty służący głównie do marnowania czasu na news'ach.
- Niezadowolenie z Minixa i rozszerzanie terminala do systemu Unixopodobnego

# Linux

- 17 września 1991 Linus Torvalds zamieszcza pierwszą wersję Linuxa w sieci.
- Hobby: poznanie architektury PC i języka C
- Terminal tty służący głównie do marnowania czasu na news'ach.
- Niezadowolenie z Minixa i rozszerzanie terminala do systemu Unixopodobnego

# Linux

- 17 września 1991 Linus Torvalds zamieszcza pierwszą wersję Linuxa w sieci.
- Hobby: poznanie architektury PC i języka C
- Terminal tty służący głównie do marnowania czasu na news'ach.
- Niezadowolenie z Minixa i rozszerzanie terminala do systemu Unixopodobnego

# Linux

- 17 września 1991 Linus Torvalds zamieszcza pierwszą wersję Linuxa w sieci.
- Hobby: poznanie architektury PC i języka C
- Terminal tty służący głównie do marnowania czasu na news'ach.
- Niezadowolenie z Minixa i rozszerzanie terminala do systemu Unixopodobnego

# Linux

- 17 września 1991 Linus Torvalds zamieszcza pierwszą wersję Linuxa w sieci.
- Hobby: poznanie architektury PC i języka C
- Terminal tty służący głównie do marnowania czasu na news'ach.
- Niezadowolenie z Minixa i rozszerzanie terminala do systemu Unixopodobnego

# GNU/Linux

- Linus od początku używał narzędzi GNU (bash, gcc)
- Działające, aktywnie rozwijane jądro było tym czego FSF brakowało.
- GNU Linux czy GNU/Linux czy Linux?
- GNU/Hurd dzisiaj



# GNU/Linux

- Linus od początku używał narzędzi GNU (bash, gcc)
- Działające, aktywnie rozwijane jądro było tym czego FSF brakowało.
- GNU Linux czy GNU/Linux czy Linux?
- GNU/Hurd dzisiaj

# GNU/Linux

- Linus od początku używał narzędzi GNU (bash, gcc)
- Działające, aktywnie rozwijane jądro było tym czego FSF brakowało.
- GNU Linux czy GNU/Linux czy Linux?
- GNU/Hurd dzisiaj

# GNU/Linux

- Linus od początku używał narzędzi GNU (bash, gcc)
- Działające, aktywnie rozwijane jądro było tym czego FSF brakowało.
- GNU Linux czy GNU/Linux czy Linux?
- GNU/Hurd dzisiaj

# GNU/Linux

- Linus od początku używał narzędzi GNU (bash, gcc)
- Działające, aktywnie rozwijane jądro było tym czego FSF brakowało.
- GNU Linux czy GNU/Linux czy Linux?
- GNU/Hurd dzisiaj

# Plan prezentacji

- 1 Wstęp
  - Historia GNU/Linux
  - Słów kilka o dystrybucjach
- 2 Porównanie dystrybucji
  - Osiołkowi w żłoby dano, czyli co dla kogo.
  - Systemy zarządzania pakietami
- 3 Standardy w systemach \*nix'owych
  - POSIX
  - SUS
  - LSB
  - FHS
  - Zgodność dystrybucji
- 4 Absolutne minimum

## Pierwsze dystrybucje

- MCC Interim Linux - Uniwersytet w Manchester (luty 1992)
- MJ - Martin Junius (lipiec 1992)
- TAMU - Uniwersytet TAMU w Teksasie (lipiec 1992)
- SLS - Softlanding Linux System (sierpień 1992)

## SLS

- Najbardziej zbliżona do dzisiejszych
- Pakiety z oprogramowaniem do wyboru:
  - podstawowy system
  - dodatki (man, emacs)
  - X-Window
  - kompilator gcc
  - T<sub>E</sub>X
  - źródła

## Powstanie dzisiejszych dystrybucji

- Slackware - Peter MacDonald (1993) na bazie SLS
- Debian - Ian Murdock (1993) comp.os.linux.development na bazie SLS
- RedHat - Marc Ewing (1994) komercyjna



# Dystrybucje dzisiaj

## Typowy skład

- Jądro systemu
- Oprogramowanie open source i/lub zamknięte
- Dokumentacja
- Multimedia

# Dystrybucje dzisiaj

## Źródła czy pakiety binarne

- "binarki" - większość
- dystrybucje typu self-hosting – źródła (np. Gentoo)

# Dystrybucje dzisiaj

## Oprogramowanie i konfiguracja

- zarządzanie pakietami
- z góry ustalona konfiguracja
- ewentualnie różne warianty (Workstation, Server, Desktop)
- administracja jest wymagana

# Plan prezentacji

- 1 Wstęp
  - Historia GNU/Linux
  - Słów kilka o dystrybucjach
- 2 Porównanie dystrybucji
  - Osiołkowi w żłoby dano, czyli co dla kogo.
  - Systemy zarządzania pakietami
- 3 Standardy w systemach \*nix'owych
  - POSIX
  - SUS
  - LSB
  - FHS
  - Zgodność dystrybucji
- 4 Absolutne minimum

## Wybór dystrybucji

- <http://eedok.voidofmind.com/linux/chooser.html>
- <http://jakilinux.org/jaki-linux/quiz>
- <http://www.zegeniestudios.net/ldc/index.php?firsttime=true>

## Pozorna klęska urodzaju

Niewiele oryginalnych dystrybucji na rynku. Większość to pochodne jednej z głównych "gałęzi".

- 129 dystrybucji bazowanych na Debianie
- 63 dystrybucje bazowane na FC
- 50 bazowanych na Knoppixie
- 28 niezależnych dystrybucji:

ALT Linux • Arch Linux • Ark Linux • CCux Linux • CRUX • Debian GNU/Linux • dyne:bolic

• Fedora Core • Freepia • Gentoo Linux • GoboLinux • Linux From Scratch • Litrix •

Mandrakelinux • Momonga Linux • Nitix Autonomic Linux • Octoz GNU/Linux • Onebase

Linux • Peanut Linux • Project dEv • Puppy Linux • QiLinux • ROCK Linux • Server optimized

Linux • Slackware Linux • Sorcerer • Specifix Linux • SUSE LINUX • UHU-Linux

- 28 bazowanych na Slackware
- 14 na RH i Mandrake
- 12 na Gentoo

# Plan prezentacji

- 1 Wstęp
  - Historia GNU/Linux
  - Słów kilka o dystrybucjach
- 2 Porównanie dystrybucji
  - Osiótkowi w żłoby dano, czyli co dla kogo.
  - Systemy zarządzania pakietami
- 3 Standardy w systemach \*nix'owych
  - POSIX
  - SUS
  - LSB
  - FHS
  - Zgodność dystrybucji
- 4 Absolutne minimum

## Ogólnie

- Linux, Unix == małe, wyspecjalizowane narzędzia
- Duże aplikacje == wiele niewielkich, kooperujących programów
- Mnogość wersji, zależności, różnice w konfiguracjach, niechciane pliki i inne problemy systemu bez pakietów.
  - Weryfikacja pakietu
  - "Prosta" instalacja, usunięcie, aktualizowanie pakietu
  - Śledzenie zależności między pakietami
  - Wprowadzanie poprawek bezpieczeństwa, uaktualnianie całego systemu.
  - Grupowanie pakietów w kategorie.



## dpkg

- dpkg to podstawowy systemem zarządzania pakietami dystrybucji systemu operacyjnego GNU/Linux Debian
- Zaprojektował i stworzył go Ian Jackson w 1993 roku.
- Narzędzie niskiego poziomu operujące na pakietach

## dpkg-APT

- APT (czyli "Advanced Packaging Tool")
- APT nie jest programem samym w sobie. Jest biblioteką języka C++ używaną przez oddzielne programy
- apt-get install php4, apt-get update, apt-get upgrade
- ściąganie pakietu (z Internetu, sieci lokalnej lub płyty CD-ROM) konfigurację, ewentualną kompilację i instalację
- dowolna liczba repozytoriów może być wpisana do /etc/apt/sources.list
- automatycznie zajmuje się zależnościami danej instalacji, proponuje zalecane pakiety, uaktualnia zainstalowane, informuje o miejscu jakie zostanie zajęte itd. itp.
- nakładki: aptitude, synaptic package manager

## src/portage/emerge

- system zarządzania pakietami napisany w pythonie wzorowany na portach z FreeBSD
- znacząco różny od dpkg czy rpm
- ebuild - plik tekstowy określający co należy poprawić, skompilować i zainstalować aby dany "pakiet" znalazł się w systemie
- emerge-sync - synchronizacja z oficjalnym drzewem Portage
- flagi USE - konfiguracja instalacji, kompilacji itd. Wpływają na zależności.
- .tbz2 - prekompilowane pakiety dla Portage. np. KDE

## rpm

- RPM (RPM Package Manager, dawniej Red Hat Package Manager)
- skompresowane bzipem2 archiwum tar z plikiem .spec
- ujednoczenie nazw - plik-wersja-wersjapakietu.architektura.rpm

## rpm-urpmi

urpmi jest to system zarządzania pakietami stworzony na potrzeby dystrybucji linuxa Mandrake Linux (przemianowanej później na Mandrakelinux, a obecnie Mandriva), a stworzony przez firmę MandrakeSoft. System urpmi wykorzystuje pakiety RPM (Red Hat Package Manager). W skład narzędzi urpmi wchodzi:

- urpme - deinstaluje wybrane pakiety wraz z zależnościami
- urpmf - wyszukuje pakiety o zadanej zawartości
- urpmi - instaluje wybrane pakiety
- urpmq - pozwala na wyświetlenie zawartości bazy pakietów
- urpmi.(addmedia,removemedi) - pozwala dodawać/usuwać źródła pakietów
- urpmi.update - aktualizuje wybrane pakiety

## rpm-yum

- Yellow Dog Updater, Modified (Fedora Core, Yellow Dog Linux)
- mniejszy od apt-rpm'a
- wolniejsze działanie
- obciążenie łącza internetowego
- brak oficjalnego GUI (wczesne prace nad pup)

## rpm-poldek

- program Pawła Gajdy (część instalatora PLD)
- nie przywiązany do jednego repozytorium (jak apt)
- łatwo decydować kiedy ma być uaktualniana lista pakietów (w przeciwieństwie do yum'a)
- ukrywanie pakietów
- możliwość łatwej wymiany programów służących do pobierania pakietów (np. wget'a)
- tryb wsadowy i interaktywny
- rozwiązywanie zależności
- mało znany

- minimalistyczne podejście do zarządzania pakietami
- brak śledzenia zależności
- tgz rozpakowany w katalogu root umieszcza pliki tam gdzie być powinny
- skrypt doinst.sh jako zaawansowane narzędzie instalacyjne ;)
- nakładki: slapt-get - udaje apt-geta, mało popularny
- stratdate: ułatwia pobieranie przy pomocy rsync gałędzi current slacka
- swaret: zarządzanie zależnościami - ldd, not found, libraries-VERSION,
- slaptpkg: skrypt basha, instalowanie i uaktualnianie pakietów z Internetu, sieci lokalnej



# LSB

- brak standardu pakietów
- rpm zamiast deb
- nie wprowadzone pomysły z portage,
- Debian podnosi rękawicę

## Popularność

- DEB - 129 dystrybucji (Debian, Knoppix, Ubuntu...)
- RPM - 112 dystrybucji (Fedora Core, PLD, Mandriva, SUSE...)
- TGZ - 39 (Slackware i pochodne)
- SRC/Portage - 10/10 (Gentoo)

# Plan prezentacji

- 1 Wstęp
  - Historia GNU/Linux
  - Słów kilka o dystrybucjach
- 2 Porównanie dystrybucji
  - Osiołkowi w żłoby dano, czyli co dla kogo.
  - Systemy zarządzania pakietami
- 3 Standardy w systemach \*nix'owych
  - POSIX
  - SUS
  - LSB
  - FHS
  - Zgodność dystrybucji
- 4 Absolutne minimum

# POSIX

- wzorcowe API dla UNIX'ów opracowane w 1985 przez IEEE (IEEE 1003)
- wysokie opłaty, brak zgody na publikację -> SUS
- drogie testy PCTS (ang. Posix Conformance Test Suite) (dalej aktualne)
- zgodność Windows z częściami POSIX'a + Cygwin
- Open Posix Test Suite <http://posixtest.sourceforge.net/>

# POSIX

- POSIX.1, System bazowy
  - Tworzenie i zarządzanie procesami
  - Sygnały
  - Wyjątki operacji zmiennopozycyjnych
  - Naruszenie segmentacji
  - Niepoprawne instrukcje
  - Błędy magistrali
  - Operacje na plikach i katalogach
  - Potoki
  - Standardowa biblioteka C
  - Interfejsy We/Wy i ich zarządzanie

# POSIX

- POSIX.1b, System czasu rzeczywistego
  - Szeregowanie z priorytetami
  - Sygnały czasu rzeczywistego
  - Zegary i wyzwalacze
  - Semafony
  - Wymiana komunikatów
  - Pamięć dzielona
  - Synchroniczne i asynchroniczne We/Wy
  - Blokady pamięci

# Plan prezentacji

- 1 Wstęp
  - Historia GNU/Linux
  - Słów kilka o dystrybucjach
- 2 Porównanie dystrybucji
  - Osiołkowi w żłoby dano, czyli co dla kogo.
  - Systemy zarządzania pakietami
- 3 Standardy w systemach \*nix'owych
  - POSIX
  - **SUS**
  - LSB
  - FHS
  - Zgodność dystrybucji
- 4 Absolutne minimum

# Single Unix Specification

- Common API Specification -> Spec 1170 -> Single UNIX Specification.
- Sun Microsystems, IBM, Hewlett-Packard, Novell/USL i OSF.
- Definicje Interfejsu Systemowego, Interfejsy Systemowe i Nagłówki, Polecenia i Narzędzia, Usługi Sieciowe, X/Open Curses
- Uzasadnienie
- bash, awk, echo, ed
- do przejścia testów nie jest niezbędny otwarty kod źródłowy.
- brak certyfikatów dla Linux'ów - zbyt częsta zmiana w dystrybucjach
- FreeBSD mocno zgodny ale nie certyfikowany



# Plan prezentacji

- 1 Wstęp
  - Historia GNU/Linux
  - Słów kilka o dystrybucjach
- 2 Porównanie dystrybucji
  - Osiołkowi w żłoby dano, czyli co dla kogo.
  - Systemy zarządzania pakietami
- 3 Standardy w systemach \*nix'owych
  - POSIX
  - SUS
  - **LSB**
  - FHS
  - Zgodność dystrybucji
- 4 Absolutne minimum

# Wstęp

- realizowany w ramach The Free Standard Group
- wspierany między innymi przez: RedHat, Novell, Adobe, Intel
- oparty na
  - POSIX
  - Single Unix Specification
  - inne (otwarte standardy)

## Cele

- rozwijanie i promowanie zbioru standardów zwiększających kompatybilność
- pomiędzy dystrybucjami Linuxa
- koordynowanie działań zachęcających firmy do pisania produktów dla Linuxa

## Zakres

- Executable and Linking Format (ELF)
- Podstawowe biblioteki
- Biblioteki narzędziowe
- Polecenia i narzędzia
- Środowisko wykonywania
- Inicjalizacja systemu
- Użytkownicy i grupy
- Format pakietów z oprogramowaniem

## Środowisko wykonywania

- dystrybucja musi być zgodna z FHS
- wymogi dla aplikacji (prawa do zapisu i odczytu niektórych plików i katalogów)

## Inicjalizacja systemu

- usługi
- skrypty inicjalizacyjne
- run-levels

Run-level	Opis
0	Wyłączenie komputera
1	Tryb pojedynczego użytkownika
2	Wielu użytkowników bez sieci
3	Wielu użytkowników z siecią
4	To samo co 3
5	To samo co 3, często tryb graficzny
6	Zrestartowanie komputera

## Format pakietów oprogramowania

- dystrybucja musi obsługiwać pakiety RPM
- dystrybucja nie musi używać RPM dla własnych pakietów
- dystrybucja nie musi używać programu rpm

## Certyfikaty

- proces certyfikacji przez The Open Group we współpracy z The Free Standard Group
- błędnie napisane testy + niespójna polityka zwalniania z testów



# Plan prezentacji

- 1 Wstęp
  - Historia GNU/Linux
  - Słów kilka o dystrybucjach
- 2 Porównanie dystrybucji
  - Osiołkowi w żłoby dano, czyli co dla kogo.
  - Systemy zarządzania pakietami
- 3 Standardy w systemach \*nix'owych
  - POSIX
  - SUS
  - LSB
  - FHS
  - Zgodność dystrybucji
- 4 Absolutne minimum

# Wstęp

- początek procesu standaryzacji (tylko Linux) – 1993
- Linux Filesystem Structure Standard – 1994 i później kolejne wersje
- FHS - także inne systemy unixowe - 1995

## Cele

- umożliwienie oprogramowaniu oraz użytkownikom przewidywania położenia zainstalowanych plików oraz katalogów.
- ograniczony zakres
  - lokalne położenie lokalnych plików to sprawa lokalna
  - wskazanie sytuacji, w których położenie plików musi być skoordynowane pomiędzy wieloma stronami jak lokalne sieci, dystrybucje, aplikacje, dokumentacja itp.

## Podstawowy podział plików

styczne	współdzielone	nie współdzielone
	/usr, /opt	/etc, /boot
dynamiczne	/var/mail, /var/spool/news	/var/run, /var/lock

## Główny system plików

- zawartość głównego systemu plików musi pozwalać na:
  - uruchomienie systemu
  - naprawienie systemu
  - przywrócenie systemu
- powinien być jak najmniejszy

## Wymagane katalogi

- bin - pliki wykonywalne podstawowych komend
- boot - statyczne pliki bootloadera
- dev - pliki urządzeń
- etc - pliki konfiguracyjne specyficzne dla hosta
- lib - główne współdzielone biblioteki i moduły jądra
- media - punkt montowania dysków wymiennych
- mnt - tymczasowy punkt montowania dla systemów plików
- opt - dodatkowe aplikacje
- sbin - podstawowe z punktu widzenia systemu pliki binarne
- srv - dane dla usług dostarczanych przez system (np. WWW, FTP)
- tmp - pliki tymczasowe
- usr - drugi oprócz głównego najważniejszy system plików
- var - dynamiczne (zmieniające się) dane

# Plan prezentacji

- 1 Wstęp
  - Historia GNU/Linux
  - Słów kilka o dystrybucjach
- 2 Porównanie dystrybucji
  - Osiołkowi w żłoby dano, czyli co dla kogo.
  - Systemy zarządzania pakietami
- 3 Standardy w systemach \*nix'owych
  - POSIX
  - SUS
  - LSB
  - FHS
  - Zgodność dystrybucji
- 4 Absolutne minimum

## Zgodność z POSIX i SUS

Wbrew pozorom spora ale bez certyfikatów. Wspomniany Open Posix Test Suite.



## Zgodność z LSB

Kontrowersyjna ze względu na błędne testy i różne do nich podejścia ze strony programistów (zmiana systemu pod błędny test bądź ignorowanie testu).

- Zgodne: Aurox, Debian, Fedora Core, Knoppix, Mandriva, PLD, Slackware, SUSE, Ubuntu i inne.
- Niezgodne: Gentoo, Arch

# Absolutne minimum

czyli czego Linuxowi do życia potrzeba

- faza inicjalizacji jądra
- faza uruchamiania programów (/sbin/init)

## Podstawowe pakiety

- SysVinit
- Bash
- Bzip2
- Coreutils
- Diffutils
- Findutils
- Gawk
- Grep
- Gzip
- NCurses
- Patch
- Sed
- Tar
- Util-linux
- Vim