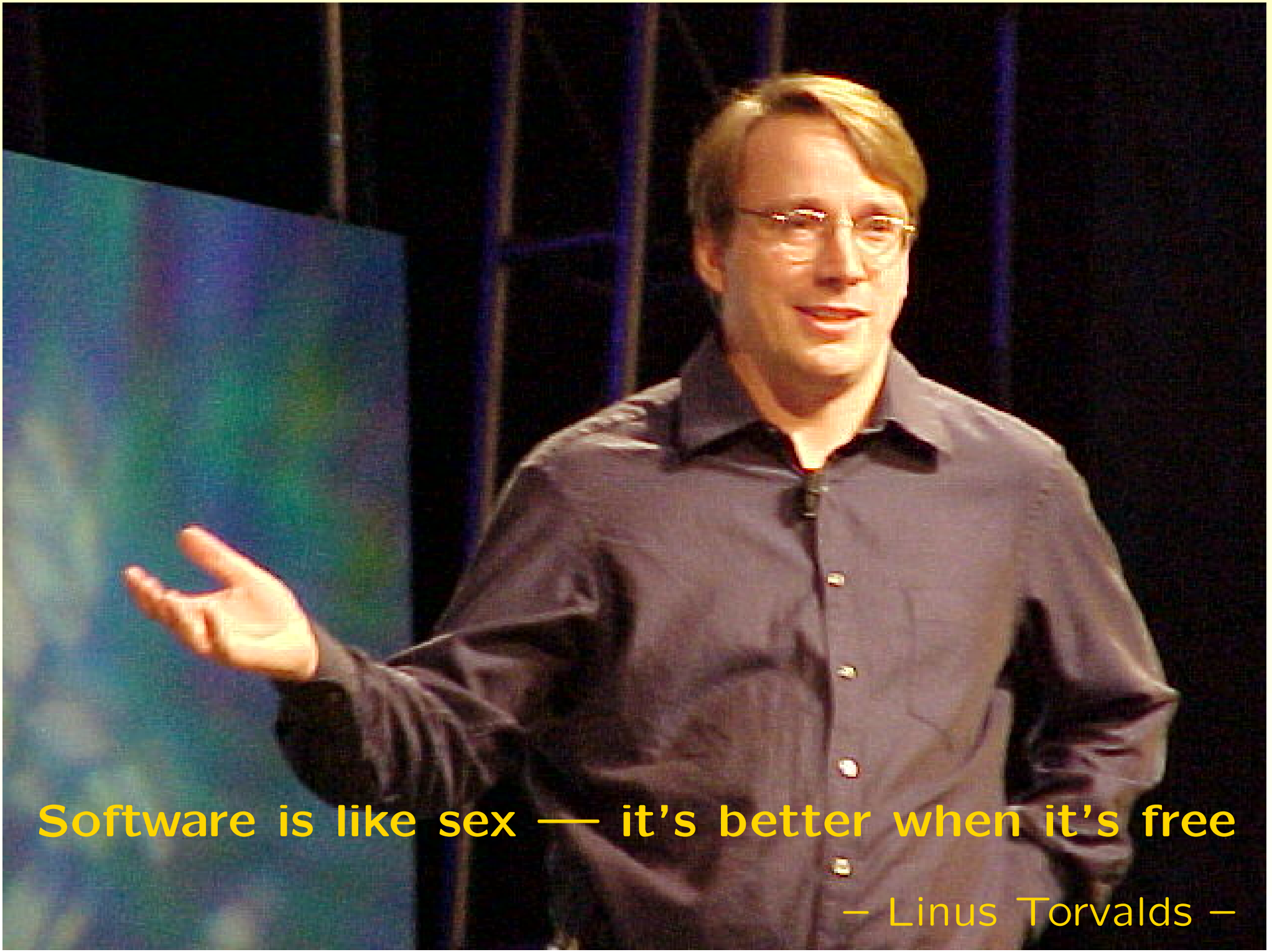

Software is like sex — it's better when it's free
– Linus Torvalds –

Free and Open Source Software

czyli

o rozwoju wolnego oprogramowania

Ryszard Tanaś

http://zon8.physd.amu.edu.pl/~tanas

13 maja 2004

Spis treści

1 Prehistoria — początki Uniksa

- 1969, Ken Thompson, AT&T Bell Labs, „niewielki ale wydajny system operacyjny z przejrzystym interfejsem usługowym”
- 1969, powstaje ARPANET
- 1977, Berkeley, pierwsza wersja BSD
- 1980, implementacja TCP/IP, wybór padł na Berkley Unix bo kod źródłowy systemu był dostępny i niezastrzeżony
- później Unix staje się produktem komercyjnym i jego źródła przestają być dostępne

1 Prehistoria — początki Uniksa

- 1969, Ken Thompson, AT&T Bell Labs, „niewielki ale wydajny system operacyjny z przejrzystym interfejsem usługowym”
- 1969, powstaje ARPANET
- 1977, Berkeley, pierwsza wersja BSD
- 1980, implementacja TCP/IP, wybór padł na Berkley Unix bo kod źródłowy systemu był dostępny i niezastrzeżony
- później Unix staje się produktem komercyjnym i jego źródła przestają być dostępne

1 Prehistoria — początki Uniksa

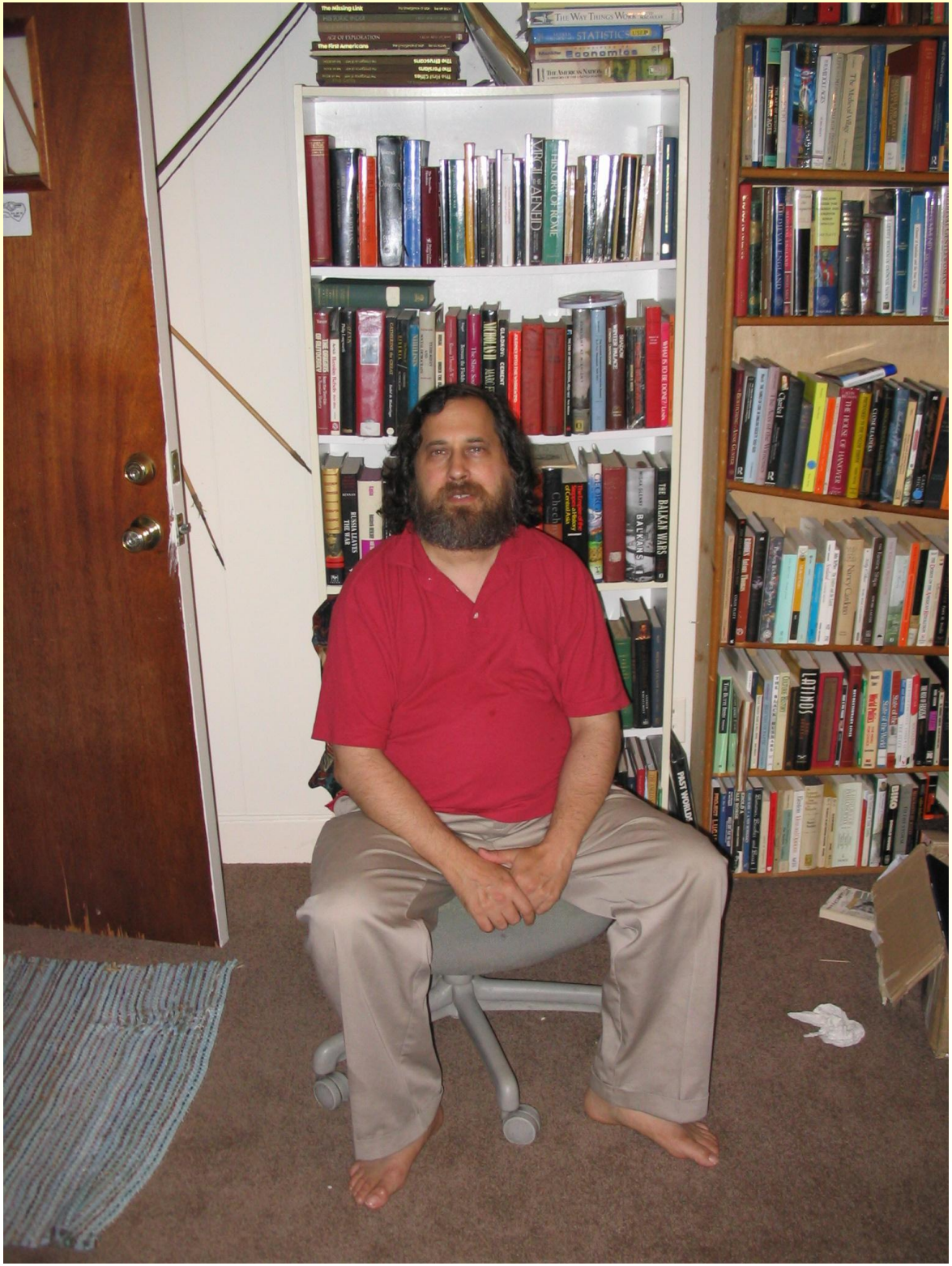
- 1969, Ken Thompson, AT&T Bell Labs, „niewielki ale wydajny system operacyjny z przejrzystym interfejsem usługowym”
- 1969, powstaje ARPANET
- 1977, Berkeley, pierwsza wersja BSD
- 1980, implementacja TCP/IP, wybór padł na Berkley Unix bo kod źródłowy systemu był dostępny i niezastrzeżony
- później Unix staje się produktem komercyjnym i jego źródła przestają być dostępne

1 Prehistoria — początki Uniksa

- 1969, Ken Thompson, AT&T Bell Labs, „niewielki ale wydajny system operacyjny z przejrzystym interfejsem usługowym”
- 1969, powstaje ARPANET
- 1977, Berkeley, pierwsza wersja BSD
- 1980, implementacja TCP/IP, wybór padł na Berkley Unix bo kod źródłowy systemu był dostępny i niezastrzeżony
- później Unix staje się produktem komercyjnym i jego źródła przestają być dostępne

1 Prehistoria — początki Uniksa

- 1969, Ken Thompson, AT&T Bell Labs, „niewielki ale wydajny system operacyjny z przejrzystym interfejsem usługowym”
- 1969, powstaje ARPANET
- 1977, Berkeley, pierwsza wersja BSD
- 1980, implementacja TCP/IP, wybór padł na Berkley Unix bo kod źródłowy systemu był dostępny i niezastrzeżony
- później Unix staje się produktem komercyjnym i jego źródła przestają być dostępne



Richard M. Stallman

ogłasza w 1985 r.

Manifest GNU

i zakłada

Free Software

Foundation

2 Manifest GNU

GNU Operating System - Free Software Foundation

Free as in Freedom

Welcome to the GNU Project web server, http://www.gnu.org. The GNU Project was launched in 1984 to develop a complete UNIX style operating system which is free software: the GNU system. (GNU is a recursive acronym for GNU's Not UNIX; it is pronounced "guh-noo".) Variants of the GNU

operating system, which use the kernel Linux, are now widely used; though these systems are often referred to as Linux, they are more accurately called GNU/Linux systems.

Czym jest Wolne Oprogramowanie?

Wolne oprogramowanie to kwestia wolności, nie ceny.
By zrozumieć tę koncepcję, powinniśmy myśleć o
wolności słowa, a nie darmowym piwie
(angielskie free znaczy najczęściej wolny, swobodny,
ale może też oznaczać darmowy).

Czym jest Wolne Oprogramowanie?

Wolne oprogramowanie to kwestia wolności, nie ceny. By zrozumieć tę koncepcję, powinniśmy myśleć o wolności słowa, a nie darmowym piwie (angielskie free znaczy najczęściej wolny, swobodny, ale może też oznaczać darmowy).

Wolne oprogramowanie odnosi się do prawa użytkowników do swobodnego uruchamiania, kopiowania, rozpowszechniania, analizowania, zmian i ulepszania programów.

Czym jest Wolne Oprogramowanie?

Wolne oprogramowanie to kwestia wolności, nie ceny. By zrozumieć tę koncepcję, powinniśmy myśleć o wolności słowa, a nie darmowym piwie (angielskie free znaczy najczęściej wolny, swobodny, ale może też oznaczać darmowy).

Wolne oprogramowanie odnosi się do prawa użytkowników do swobodnego uruchamiania, kopiowania, rozpowszechniania, analizowania, zmian i ulepszania programów.

Dokładniej, mówimy o czterech rodzajach wolności użytkowników programu:

- wolność uruchamiania programu, w dowolnym celu (wolność 0),
- wolność analizowania, jak program działa, i dostosowywania go do swoich potrzeb (wolność 1). Warunkiem koniecznym jest tu dostęp do kodu źródłowego.
- wolność rozpowszechniania kopii, byście mogli pomóc sąsiadom (wolność 2)
- wolność udoskonalania programu i publicznego rozpowszechniania własnych ulepszeń, dzięki czemu może z nich skorzystać cała społeczność (wolność 3). Warunkiem koniecznym jest tu dostęp do kodu źródłowego.

- wolność uruchamiania programu, w dowolnym celu (wolność 0),
- wolność analizowania, jak program działa, i dostosowywania go do swoich potrzeb (wolność 1). Warunkiem koniecznym jest tu dostęp do kodu źródłowego.
- wolność rozpowszechniania kopii, byście mogli pomóc sąsiadom (wolność 2)
- wolność udoskonalania programu i publicznego rozpowszechniania własnych ulepszeń, dzięki czemu może z nich skorzystać cała społeczność (wolność 3). Warunkiem koniecznym jest tu dostęp do kodu źródłowego.

- wolność uruchamiania programu, w dowolnym celu (wolność 0),
- wolność analizowania, jak program działa, i dostosowywania go do swoich potrzeb (wolność 1). Warunkiem koniecznym jest tu dostęp do kodu źródłowego.
- wolność rozpowszechniania kopii, byście mogli pomóc sąsiadom (wolność 2)
- wolność udoskonalania programu i publicznego rozpowszechniania własnych ulepszeń, dzięki czemu może z nich skorzystać cała społeczność (wolność 3). Warunkiem koniecznym jest tu dostęp do kodu źródłowego.

- wolność uruchamiania programu, w dowolnym celu (wolność 0),
- wolność analizowania, jak program działa, i dostosowywania go do swoich potrzeb (wolność 1). Warunkiem koniecznym jest tu dostęp do kodu źródłowego.
- wolność rozpowszechniania kopii, byście mogli pomóc sąsiadom (wolność 2)
- wolność udoskonalania programu i publicznego rozpowszechniania własnych ulepszeń, dzięki czemu może z nich skorzystać cała społeczność (wolność 3). Warunkiem koniecznym jest tu dostęp do kodu źródłowego.

3 Kategorie oprogramowania

- Wolne oprogramowanie (free software)
- Oprogramowanie z udostępnionym kodem źródłowym (open source)
- Oprogramowanie będące dobrem publicznym (public domain)
- Oprogramowanie objęte copyleft
- Wolne oprogramowanie nie objęte copyleft
- Oprogramowanie objęte GPL
- Oprogramowanie GNU
- Oprogramowanie półwolne (semi-free software)

3 Kategorie oprogramowania

- Wolne oprogramowanie (free software)
- Oprogramowanie z udostępnionym kodem źródłowym (open source)
- Oprogramowanie będące dobrem publicznym (public domain)
- Oprogramowanie objęte copyleft
- Wolne oprogramowanie nie objęte copyleft
- Oprogramowanie objęte GPL
- Oprogramowanie GNU
- Oprogramowanie półwolne (semi-free software)

3 Kategorie oprogramowania

- Wolne oprogramowanie (free software)
- Oprogramowanie z udostępnionym kodem źródłowym (open source)
- **Oprogramowanie będące dobrem publicznym (public domain)**
- Oprogramowanie objęte copyleft
- Wolne oprogramowanie nie objęte copyleft
- Oprogramowanie objęte GPL
- Oprogramowanie GNU
- Oprogramowanie półwolne (semi-free software)

3 Kategorie oprogramowania

- Wolne oprogramowanie (free software)
- Oprogramowanie z udostępnionym kodem źródłowym (open source)
- Oprogramowanie będące dobrem publicznym (public domain)
- Oprogramowanie objęte copyleft
- Wolne oprogramowanie nie objęte copyleft
- Oprogramowanie objęte GPL
- Oprogramowanie GNU
- Oprogramowanie półwolne (semi-free software)

3 Kategorie oprogramowania

- Wolne oprogramowanie (free software)
- Oprogramowanie z udostępnionym kodem źródłowym (open source)
- Oprogramowanie będące dobrem publicznym (public domain)
- Oprogramowanie objęte copyleft
- Wolne oprogramowanie nie objęte copyleft
- Oprogramowanie objęte GPL
- Oprogramowanie GNU
- Oprogramowanie półwolne (semi-free software)

3 Kategorie oprogramowania

- Wolne oprogramowanie (free software)
- Oprogramowanie z udostępnionym kodem źródłowym (open source)
- Oprogramowanie będące dobrem publicznym (public domain)
- Oprogramowanie objęte copyleft
- Wolne oprogramowanie nie objęte copyleft
- Oprogramowanie objęte GPL
- Oprogramowanie GNU
- Oprogramowanie półwolne (semi-free software)

3 Kategorie oprogramowania

- Wolne oprogramowanie (free software)
- Oprogramowanie z udostępnionym kodem źródłowym (open source)
- Oprogramowanie będące dobrem publicznym (public domain)
- Oprogramowanie objęte copyleft
- Wolne oprogramowanie nie objęte copyleft
- Oprogramowanie objęte GPL
- Oprogramowanie GNU
- Oprogramowanie półwolne (semi-free software)

3 Kategorie oprogramowania

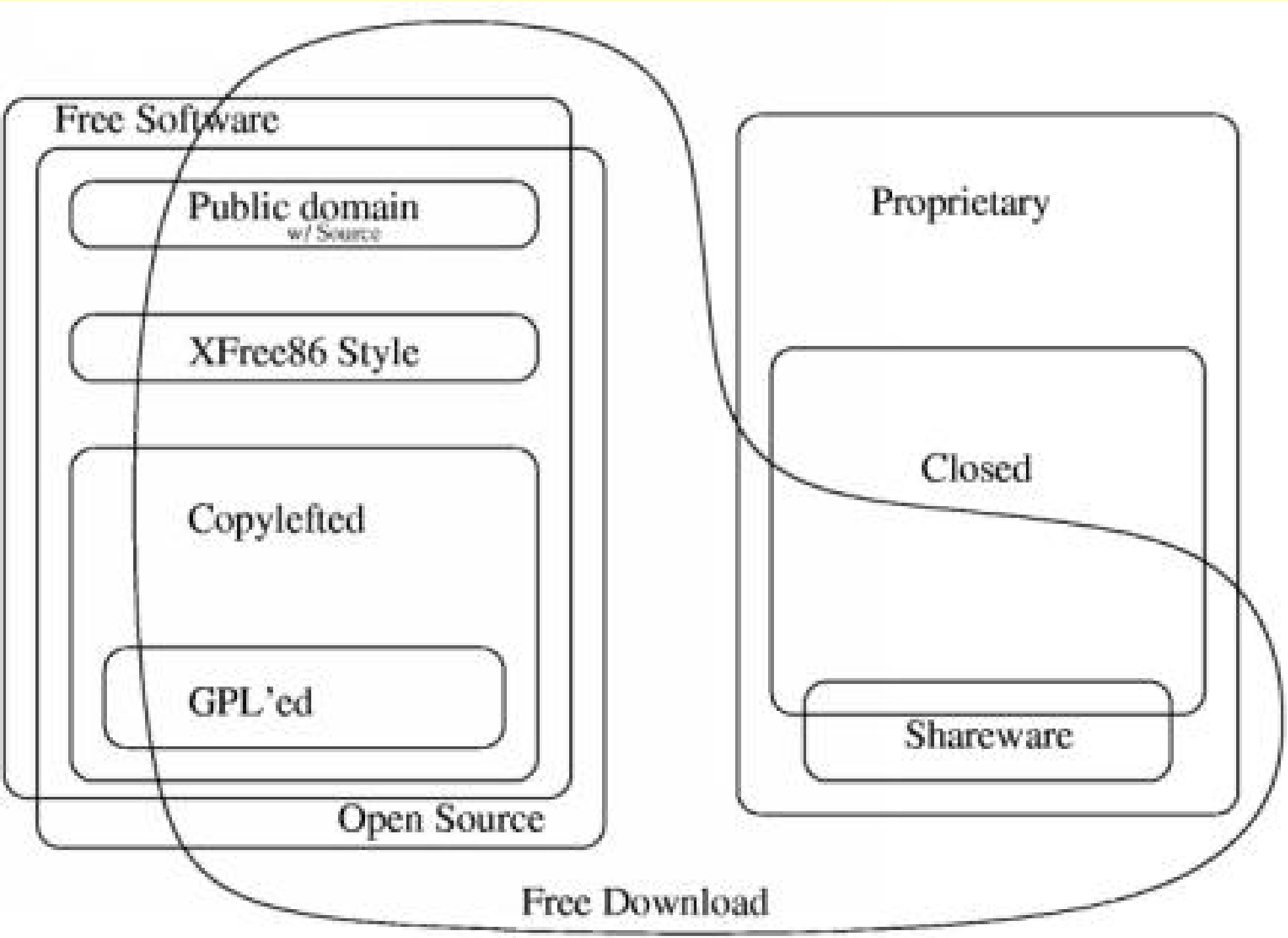
- Wolne oprogramowanie (free software)
- Oprogramowanie z udostępnionym kodem źródłowym (open source)
- Oprogramowanie będące dobrem publicznym (public domain)
- Oprogramowanie objęte copyleft
- Wolne oprogramowanie nie objęte copyleft
- Oprogramowanie objęte GPL
- Oprogramowanie GNU
- Oprogramowanie półwolne (semi-free software)

- Oprogramowanie prawnie zastrzeżone (proprietary software)
- Freeware
- Shareware
- Oprogramowanie komercyjne (commercial software)

- Oprogramowanie prawnie zastrzeżone (proprietary software)
- Freeware
- Shareware
- Oprogramowanie komercyjne (commercial software)

- Oprogramowanie prawnie zastrzeżone (proprietary software)
- Freeware
- Shareware
- Oprogramowanie komercyjne (commercial software)

- Oprogramowanie prawnie zastrzeżone (proprietary software)
- Freeware
- Shareware
- Oprogramowanie komercyjne (commercial software)


Free Software
Public domain
w/ Source
XFree86 Style
Copylefted
GPL'ed
Open Source
Proprietary
Closed
Shareware
Free Download

4 Licencja GPL

Licencja GPL (General Public License) jest najczęściej stosowaną licencją otwartego oprogramowania.
Licencja GPL wymaga aby każdy program zawierający części objęte tą licencją sam był w całości rozprowadzany zgodnie z jej zasadami.

4 Licencja GPL

Licencja GPL (General Public License) jest najczęściej stosowaną licencją otwartego oprogramowania.
Licencja GPL wymaga aby każdy program zawierający części objęte tą licencją sam był w całości rozprowadzany zgodnie z jej zasadami.

Istnieją mniej restrykcyjne licencje oprogramowania otwartego, takie jak: Licencja MIT lub X Consortium, licencja BSD, licencja Artistic czy Mozilla Public License.

5 Linux

W 1991 r. Linus Torvalds, student Uniwersytetu w Helsinkach, korzystając z systemu Minix jako wzorca, stworzył jądro systemu operacyjnego Linux (nazwa pochodzi od Linux Is Not UniX).

5 Linux

W 1991 r. Linus Torvalds, student Uniwersytetu w Helsinkach, korzystając z systemu Minix jako wzorca, stworzył jądro systemu operacyjnego Linux (nazwa pochodzi od Linux Is Not UniX).

Tego właśnie brakowało w systemie GNU!

5 Linux

W 1991 r. Linus Torvalds, student Uniwersytetu w Helsinkach, korzystając z systemu Minix jako wzorca, stworzył jądro systemu operacyjnego Linux (nazwa pochodzi od Linux Is Not UniX).

Tego właśnie brakowało w systemie GNU!

Połączenie jądra Linuksa z istniejącymi aplikacjami GNU dało system GNU/Linux i zapoczątkowało jego burzliwy rozwój, który stał się symbolem i fenomenem wolnego oprogramowania.

5 Linux

W 1991 r. Linus Torvalds, student Uniwersytetu w Helsinkach, korzystając z systemu Minix jako wzorca, stworzył jądro systemu operacyjnego Linux (nazwa pochodzi od Linux Is Not UniX).

Tego właśnie brakowało w systemie GNU!

Połączenie jądra Linuksa z istniejącymi aplikacjami GNU dało system GNU/Linux i zapoczątkowało jego burzliwy rozwój, który stał się symbolem i fenomenem wolnego oprogramowania.

Do rozwoju tego przyczyniła się eksplozja internetu w latach 1993-1994.

6 Open source — oprogramowanie otwarte

Eric S. Raymond
tworzy w 1998 r.
Open Source Initiative
autor
The Cathedral and the Bazaar

Bardziej pragmatyczne podejście do problemu wolnego oprogramowania. Obecnie mówi się o Free and Open Source Software (FOSS).

Istotne momenty w rozwoju FOSS

- 1994, powstaje Red Hat
- 1996, startuje projekt KDE
- 1997, startuje projekt GNOME
- 1998, powstaje Open Source Initiative
- 1998, Netscape udostępnia źródła przeglądarki Netscape Navigator
- 2000, Sun uwalnia źródła Star Office
- 2002, pojawia się wersja 1.0 przeglądarki Mozilla

Istotne momenty w rozwoju FOSS

- 1994, powstaje Red Hat
- 1996, startuje projekt KDE
- 1997, startuje projekt GNOME
- 1998, powstaje Open Source Initiative
- 1998, Netscape udostępnia źródła przeglądarki Netscape Navigator
- 2000, Sun uwalnia źródła Star Office
- 2002, pojawia się wersja 1.0 przeglądarki Mozilla

Istotne momenty w rozwoju FOSS

- 1994, powstaje Red Hat
- 1996, startuje projekt KDE
- 1997, startuje projekt GNOME
- 1998, powstaje Open Source Initiative
- 1998, Netscape udostępnia źródła przeglądarki Netscape Navigator
- 2000, Sun uwalnia źródła Star Office
- 2002, pojawia się wersja 1.0 przeglądarki Mozilla

Istotne momenty w rozwoju FOSS

- 1994, powstaje Red Hat
- 1996, startuje projekt KDE
- 1997, startuje projekt GNOME
- 1998, powstaje Open Source Initiative
- 1998, Netscape udostępnia źródła przeglądarki Netscape Navigator
- 2000, Sun uwalnia źródła Star Office
- 2002, pojawia się wersja 1.0 przeglądarki Mozilla

Istotne momenty w rozwoju FOSS

- 1994, powstaje Red Hat
- 1996, startuje projekt KDE
- 1997, startuje projekt GNOME
- 1998, powstaje Open Source Initiative
- 1998, Netscape udostępnia źródła przeglądarki Netscape Navigator
- 2000, Sun uwalnia źródła Star Office
- 2002, pojawia się wersja 1.0 przeglądarki Mozilla

Istotne momenty w rozwoju FOSS

- 1994, powstaje Red Hat
- 1996, startuje projekt KDE
- 1997, startuje projekt GNOME
- 1998, powstaje Open Source Initiative
- 1998, Netscape udostępnia źródła przeglądarki Netscape Navigator
- 2000, Sun uwalnia źródła Star Office
- 2002, pojawia się wersja 1.0 przeglądarki Mozilla

Istotne momenty w rozwoju FOSS

- 1994, powstaje Red Hat
- 1996, startuje projekt KDE
- 1997, startuje projekt GNOME
- 1998, powstaje Open Source Initiative
- 1998, Netscape udostępnia źródła przeglądarki Netscape Navigator
- 2000, Sun uwalnia źródła Star Office
- 2002, pojawia się wersja 1.0 przeglądarki Mozilla

7 Najbardziej znane programy FOSS

- Systemy operacyjne: Linux, FreeBSD, OpenBSD, NetBSD, GNU/Hurd
- Języki: GNU C/C++, Perl, Python, TCL
- Edytory: Vi, Emacs
- Systemy okien: The X Window System, XFree86
- Środowiska graficzne: GNOME, KDE, GNUStep, XFce
- Przeglądarki: Mozilla, Galeon
- Aplikacje: ABIWord, GIMP
- Pakiety biurowe: Open Office, KOffice

7 Najbardziej znane programy FOSS

- Systemy operacyjne: Linux, FreeBSD, OpenBSD, NetBSD, GNU/Hurd
- Języki: GNU C/C++, Perl, Python, TCL
- Edytory: Vi, Emacs
- Systemy okien: The X Window System, XFree86
- Środowiska graficzne: GNOME, KDE, GNUStep, XFce
- Przeglądarki: Mozilla, Galeon
- Aplikacje: ABIWord, GIMP
- Pakiety biurowe: Open Office, KOffice

7 Najbardziej znane programy FOSS

- Systemy operacyjne: Linux, FreeBSD, OpenBSD, NetBSD, GNU/Hurd
- Języki: GNU C/C++, Perl, Python, TCL
- Edytory: Vi, Emacs
- Systemy okien: The X Window System, XFree86
- Środowiska graficzne: GNOME, KDE, GNUStep, XFce
- Przeglądarki: Mozilla, Galeon
- Aplikacje: ABIWord, GIMP
- Pakiety biurowe: Open Office, KOffice

7 Najbardziej znane programy FOSS

- Systemy operacyjne: Linux, FreeBSD, OpenBSD, NetBSD, GNU/Hurd
- Języki: GNU C/C++, Perl, Python, TCL
- Edytory: Vi, Emacs
- Systemy okien: The X Window System, XFree86
- Środowiska graficzne: GNOME, KDE, GNUStep, XFce
- Przeglądarki: Mozilla, Galeon
- Aplikacje: ABIWord, GIMP
- Pakiety biurowe: Open Office, KOffice

7 Najbardziej znane programy FOSS

- Systemy operacyjne: Linux, FreeBSD, OpenBSD, NetBSD, GNU/Hurd
- Języki: GNU C/C++, Perl, Python, TCL
- Edytory: Vi, Emacs
- Systemy okien: The X Window System, XFree86
- Środowiska graficzne: GNOME, KDE, GNUStep, XFce
- Przeglądarki: Mozilla, Galeon
- Aplikacje: ABIWord, GIMP
- Pakiety biurowe: Open Office, KOffice

7 Najbardziej znane programy FOSS

- Systemy operacyjne: Linux, FreeBSD, OpenBSD, NetBSD, GNU/Hurd
- Języki: GNU C/C++, Perl, Python, TCL
- Edytory: Vi, Emacs
- Systemy okien: The X Window System, XFree86
- Środowiska graficzne: GNOME, KDE, GNUStep, XFce
- Przeglądarki: Mozilla, Galeon
- Aplikacje: ABIWord, GIMP
- Pakiety biurowe: Open Office, KOffice

7 Najbardziej znane programy FOSS

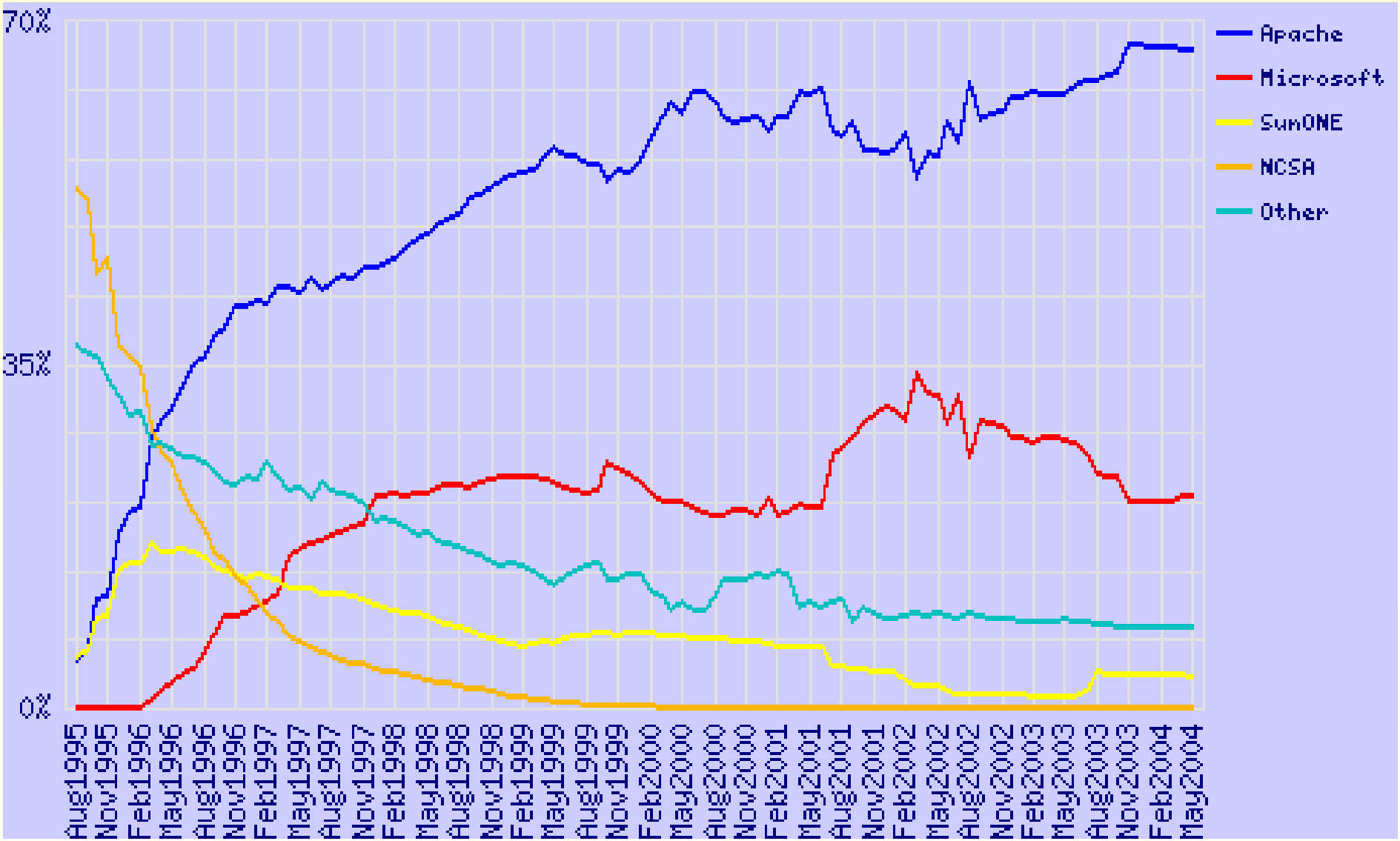
- Systemy operacyjne: Linux, FreeBSD, OpenBSD, NetBSD, GNU/Hurd
- Języki: GNU C/C++, Perl, Python, TCL
- Edytory: Vi, Emacs
- Systemy okien: The X Window System, XFree86
- Środowiska graficzne: GNOME, KDE, GNUStep, XFce
- Przeglądarki: Mozilla, Galeon
- Aplikacje: ABIWord, GIMP
- Pakiety biurowe: Open Office, KOffice

7 Najbardziej znane programy FOSS

- Systemy operacyjne: Linux, FreeBSD, OpenBSD, NetBSD, GNU/Hurd
- Języki: GNU C/C++, Perl, Python, TCL
- Edytory: Vi, Emacs
- Systemy okien: The X Window System, XFree86
- Środowiska graficzne: GNOME, KDE, GNUStep, XFce
- Przeglądarki: Mozilla, Galeon
- Aplikacje: ABIWord, GIMP
- Pakiety biurowe: Open Office, KOffice

- Oprogramowanie serwerowe: Apache, Samba, PHP, MySQL, PostgerSQL

- Oprogramowanie serwerowe: Apache, Samba, PHP, MySQL, PostgerSQL



Serwery WWW

8 Tworzenie oprogramowania — wersja „katedralna”

- Analiza wymagań rynku
- Projekt ogólny, na poziomie systemowym
- Projekt szczegółowy, modularyzacja
- Implementacja
- Integracja
- Testowanie
- Wsparcie techniczne

8 Tworzenie oprogramowania — wersja „katedralna”

- Analiza wymagań rynku
- Projekt ogólny, na poziomie systemowym
- Projekt szczegółowy, modularyzacja
- Implementacja
- Integracja
- Testowanie
- Wsparcie techniczne

8 Tworzenie oprogramowania — wersja „katedralna”

- Analiza wymagań rynku
- Projekt ogólny, na poziomie systemowym
- Projekt szczegółowy, modularyzacja
- Implementacja
- Integracja
- Testowanie
- Wsparcie techniczne

8 Tworzenie oprogramowania — wersja „katedralna”

- Analiza wymagań rynku
- Projekt ogólny, na poziomie systemowym
- Projekt szczegółowy, modularyzacja
- Implementacja
- Integracja
- Testowanie
- Wsparcie techniczne

8 Tworzenie oprogramowania — wersja „katedralna”

- Analiza wymagań rynku
- Projekt ogólny, na poziomie systemowym
- Projekt szczegółowy, modularyzacja
- Implementacja
- Integracja
- Testowanie
- Wsparcie techniczne

8 Tworzenie oprogramowania — wersja „katedralna”

- Analiza wymagań rynku
- Projekt ogólny, na poziomie systemowym
- Projekt szczegółowy, modularyzacja
- Implementacja
- Integracja
- Testowanie
- Wsparcie techniczne

8 Tworzenie oprogramowania — wersja „katedralna”

- Analiza wymagań rynku
- Projekt ogólny, na poziomie systemowym
- Projekt szczegółowy, modularyzacja
- Implementacja
- Integracja
- Testowanie
- Wsparcie techniczne

8 Tworzenie oprogramowania — wersja „katedralna”

- Analiza wymagań rynku
- Projekt ogólny, na poziomie systemowym
- Projekt szczegółowy, modularyzacja
- Implementacja
- Integracja
- Testowanie
- Wsparcie techniczne

Projekt tworzony przez zamkniętą grupę programistów. Kolejne wersje wydawane rzadko po „należytym” przetestowaniu.

9 Tworzenie oprogramowania — model „bazarowy”

- Otwartość źródeł (wielu ma dostęp)
- Programy należy udostępniać szybko i często (wielu testuje i poprawia)
- Zasługi współtwórców są nagradzane uznaniem

9 Tworzenie oprogramowania — model „bazarowy”

- Otwartość źródeł (wielu ma dostęp)
- Programy należy udostępniać szybko i często (wielu testuje i poprawia)
- Zasługi współtwórców są nagradzane uznaniem

9 Tworzenie oprogramowania — model „bazarowy”

- Otwartość źródeł (wielu ma dostęp)
- Programy należy udostępniać szybko i często (wielu testuje i poprawia)
- Zasługi współtwórców są nagradzane uznaniem

9 Tworzenie oprogramowania — model „bazarowy”

- Otwartość źródeł (wielu ma dostęp)
- Programy należy udostępniać szybko i często (wielu testuje i poprawia)
- Zasługi współtwórców są nagradzane uznaniem

Projekt tworzony przez dużą liczbę ochotników, koordynowany przez uznanego przez społeczność lidera. Kolejne wersje wydawane często (czasem kilka razy dziennie!) i testowane przez wszystkich, którzy mają na to ochotę. Każdy może dołożyć własną cegiełkę do projektu i jego wkład będzie zaznaczony i uznany.

10 Czym więc jest FOSS?

Jest to oprogramowanie dla którego:

- kod źródłowy jest dostępny dla użytkownika
- użytkownik może modyfikować kod źródłowy
- warunki licencji ułatwiają dostęp do oprogramowania i jego rozpowszechniania
- koszty nabycia są minimalne

10 Czym więc jest FOSS?

Jest to oprogramowanie dla którego:

- kod źródłowy jest dostępny dla użytkownika
- użytkownik może modyfikować kod źródłowy
- warunki licencji ułatwiają dostęp do oprogramowania i jego rozpowszechniania
- koszty nabycia są minimalne

10 Czym więc jest FOSS?

Jest to oprogramowanie dla którego:

- kod źródłowy jest dostępny dla użytkownika
- użytkownik może modyfikować kod źródłowy
- warunki licencji ułatwiają dostęp do oprogramowania i jego rozpowszechniania
- koszty nabycia są minimalne

10 Czym więc jest FOSS?

Jest to oprogramowanie dla którego:

- kod źródłowy jest dostępny dla użytkownika
- użytkownik może modyfikować kod źródłowy
- warunki licencji ułatwiają dostęp do oprogramowania i jego rozpowszechniania
- koszty nabycia są minimalne

11 Zalety FOSS

- nie ma sekretów, każdy ma wgląd w jego strukturę, algorytmy, itp
- nie ma jednego właściciela, trzyma się więc raczej otwartych niż zamkniętych standardów
- jest podtrzymywane przez społeczności a nie korporacje, poprawki są szybkie i darmowe
- jest zwykle darmowe, a twórcy zarabiają na wsparciu technicznym, szkoleniu i specjalistycznych dodatkach

11 Zalety FOSS

- nie ma sekretów, każdy ma wgląd w jego strukturę, algorytmy, itp
- nie ma jednego właściciela, trzyma się więc raczej otwartych niż zamkniętych standardów
- jest podtrzymywane przez społeczności a nie korporacje, poprawki są szybkie i darmowe
- jest zwykle darmowe, a twórcy zarabiają na wsparciu technicznym, szkoleniu i specjalistycznych dodatkach

11 Zalety FOSS

- nie ma sekretów, każdy ma wgląd w jego strukturę, algorytmy, itp
- nie ma jednego właściciela, trzyma się więc raczej otwartych niż zamkniętych standardów
- jest podtrzymywane przez społeczności a nie korporacje, poprawki są szybkie i darmowe
- jest zwykle darmowe, a twórcy zarabiają na wsparciu technicznym, szkoleniu i specjalistycznych dodatkach

11 Zalety FOSS

- nie ma sekretów, każdy ma wgląd w jego strukturę, algorytmy, itp
- nie ma jednego właściciela, trzyma się więc raczej otwartych niż zamkniętych standardów
- jest podtrzymywane przez społeczności a nie korporacje, poprawki są szybkie i darmowe
- jest zwykle darmowe, a twórcy zarabiają na wsparciu technicznym, szkoleniu i specjalistycznych dodatkach

12 Wady FOSS

- nierównomierne tempo pracy nad poszczególnymi elementami projektu
- brak zwierzchnika, który wymuszał by właściwe tempo
- zmienne zasoby ludzkie (nie ma godzin pracy!)
- czasem rozbieżne cele i aspiracje
- projekt nie musi się zakończyć sukcesem

12 Wady FOSS

- nierównomierne tempo pracy nad poszczególnymi elementami projektu
- brak zwierzchnika, który wymuszał by właściwe tempo
- zmienne zasoby ludzkie (nie ma godzin pracy!)
- czasem rozbieżne cele i aspiracje
- projekt nie musi się zakończyć sukcesem

12 Wady FOSS

- nierównomierne tempo pracy nad poszczególnymi elementami projektu
- brak zwierzchnika, który wymuszał by właściwe tempo
- zmienne zasoby ludzkie (nie ma godzin pracy!)
- czasem rozbieżne cele i aspiracje
- projekt nie musi się zakończyć sukcesem

12 Wady FOSS

- nierównomierne tempo pracy nad poszczególnymi elementami projektu
- brak zwierzchnika, który wymuszał by właściwe tempo
- zmienne zasoby ludzkie (nie ma godzin pracy!)
- czasem rozbieżne cele i aspiracje
- projekt nie musi się zakończyć sukcesem

12 Wady FOSS

- nierównomierne tempo pracy nad poszczególnymi elementami projektu
- brak zwierzchnika, który wymuszał by właściwe tempo
- zmienne zasoby ludzkie (nie ma godzin pracy!)
- czasem rozbieżne cele i aspiracje
- projekt nie musi się zakończyć sukcesem

13 Częste zarzuty choć niekoniecznie prawdziwe

- kiepska jakość
- powolny rozwój
- nie tak dopracowane jak komercyjne
- nie ma wsparcia technicznego

13 Częste zarzuty choć niekoniecznie prawdziwe

- kiepska jakość
- powolny rozwój
- nie tak dopracowane jak komercyjne
- nie ma wsparcia technicznego

13 Częste zarzuty choć niekoniecznie prawdziwe

- kiepska jakość
- powolny rozwój
- nie tak dopracowane jak komercyjne
- nie ma wsparcia technicznego

13 Częste zarzuty choć niekoniecznie prawdziwe

- kiepska jakość
- powolny rozwój
- nie tak dopracowane jak komercyjne
- nie ma wsparcia technicznego

13 Częste zarzuty choć niekoniecznie prawdziwe

- kiepska jakość
- powolny rozwój
- nie tak dopracowane jak komercyjne
- nie ma wsparcia technicznego

14 Kto pisze FOSS?

Są to zwykle młodzi ludzie, ktorzy chcą się uczyć od innych, poznawać nowe technologie i mieć kontakt z ludźmi pracującymi nad czymś ważnym. Często potem pracują nad FOSS dla określonych firm.

15 Kto za to płaci?

- ludzie, którzy sami potrzebują oprogramowania we własnym biznesie
- Linux jest wspierany przez producentów hardware'u (IBM, HP, Sun) bo zwiększa sprzedaż hardware'u
- często firmy zatrudniają programistów chociaż nie produkują oprogramowania
- koszty oprogramowania rozkładają się na wielu, koszt jednostkowy jest mniejszy
- istnieją już firmy zarabiające na FOSS

15 Kto za to płaci?

- ludzie, którzy sami potrzebują oprogramowania we własnym biznesie
- Linux jest wspierany przez producentów hardware'u (IBM, HP, Sun) bo zwiększa sprzedaż hardware'u
- często firmy zatrudniają programistów chociaż nie produkują oprogramowania
- koszty oprogramowania rozkładają się na wielu, koszt jednostkowy jest mniejszy
- istnieją już firmy zarabiające na FOSS

15 Kto za to płaci?

- ludzie, którzy sami potrzebują oprogramowania we własnym biznesie
- Linux jest wspierany przez producentów hardware'u (IBM, HP, Sun) bo zwiększa sprzedaż hardware'u
- często firmy zatrudniają programistów chociaż nie produkują oprogramowania
- koszty oprogramowania rozkładają się na wielu, koszt jednostkowy jest mniejszy
- istnieją już firmy zarabiające na FOSS

15 Kto za to płaci?

- ludzie, którzy sami potrzebują oprogramowania we własnym biznesie
- Linux jest wspierany przez producentów hardware'u (IBM, HP, Sun) bo zwiększa sprzedaż hardware'u
- często firmy zatrudniają programistów chociaż nie produkują oprogramowania
- koszty oprogramowania rozkładają się na wielu, koszt jednostkowy jest mniejszy
- istnieją już firmy zarabiające na FOSS

15 Kto za to płaci?

- ludzie, którzy sami potrzebują oprogramowania we własnym biznesie
- Linux jest wspierany przez producentów hardware'u (IBM, HP, Sun) bo zwiększa sprzedaż hardware'u
- często firmy zatrudniają programistów chociaż nie produkują oprogramowania
- koszty oprogramowania rozkładają się na wielu, koszt jednostkowy jest mniejszy
- istnieją już firmy zarabiające na FOSS

Good programmers know what to write.

Great ones know what to rewrite (and reuse).

– Eric Raymond –

Good programmers know what to write.

Great ones know what to rewrite (and reuse).

– Eric Raymond –

Powodzenia!