

BSD – alternatywa dla Linuksa

(na przykładzie FreeBSD)





Plan prezentacji

1. Historia BSD



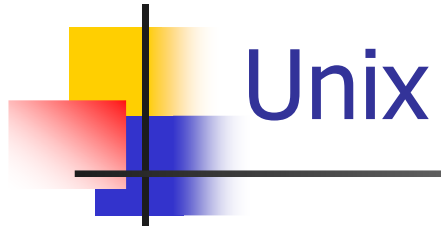
Plan prezentacji

1. Historia BSD
2. Licencja BSD a GNU/GPL

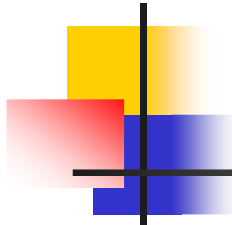


Plan prezentacji

1. Historia BSD
2. Licencja BSD a GNU/GPL
3. Brak dystrybucji – wada czy zaleta ?



- **1969:** AT&T Bell Labs (K. Thompson, D. Ritchie) [asm na PDP-7]

The logo consists of a vertical black line intersecting a horizontal black line. To the left of the intersection, there are three overlapping squares: a yellow one at the top, a red one to the left, and a blue one at the bottom. The word "Unix" is written in a blue, sans-serif font to the right of the vertical line.

Unix

- **1969:** AT&T Bell Labs (K. Thompson, D. Ritchie) [asm na PDP-7]
- **1973:** *Version 4* - przepisane w C, udostępnienie kodu (oprócz jądra) [PDP-11]
- **1975:** *Version 6*

Berkeley Software Distribution (Berkeley Unix)



Bill Joy (* 1954)

Berkeley Software Distribution (Berkeley Unix)



Bill Joy (* 1954)

- **1977:** 1BSD (*kompilator
Pascala, ex*)
- **1978:** 2BSD (*vi, termcap, csh*)
[PDP -> VAX]

Berkeley Software Distribution (Berkeley Unix)



Bill Joy (* 1954)

- **1977:** 1BSD (*kompilator Pascala, ex*)
- **1978:** 2BSD (*vi, termcap, csh*)
[PDP -> VAX]
- **1979:** 3BSD (popularność)
Defense Advanced Research
Projects Agency (DARPA)
funduje Computer Systems
Research Group (CSRG) [„VLSI
Project”]



Lata 80.

- **1980:** CSRG tworzy 4BSD
- **1981:** 4.1 BSD



Lata 80.

- **1980:** CSRG tworzy 4BSD
- **1981:** 4.1 BSD
IBM-PC
- **1982:** AT&T – Unix System III (komercyjny)
MS-DOS 1.25
- **1983:** GNU (R.Stallman)





Lata 80.

- **1980:** CSRG tworzy 4BSD
- **1981:** 4.1 BSD
IBM-PC
- **1982:** GATT – Unix System III (komercyjny)
MS-DOS 1.25
- **1983:** GNU (R.Stallman)
- **1984:** 4.2 BSD (*TCP/IP*) [1000 sztuk !]
- **1985-93:** „Unix wars” (AT&T+ Sun vs. OSF)
- **1986:** 4.3 BSD (*demon DNS*)
- **1988:** BSD – [VAX -> inne maszyny]



Lata 90.

- **1991:** Linux
- **1992:** Proces AT&T (USL) vs. Berkeley Software Design (BSDi) o BSD/386 [komercyjny BSD na i386] (zahamowanie 386BSD, NetBSD, FreeBSD)
- **1993:** 4.4BSD
FreeBSD, NetBSD



Lata 90.

- **1991:** Linux
- **1992:** Proces AT&T (USL) vs. Berkeley Software Design (BSDi) o BSD/386 [komercyjny BSD na i386] (zahamowanie 386BSD, NetBSD, FreeBSD)
- **1993:** 4.4BSD
FreeBSD, NetBSD
- **1994:** ugoda – 4.4BSD.Lite (wolny od roszczeń)
- **1995:** 4.4BSD Lite2 – podstawa dla FreeBSD (3.0), NetBSD (1.3) oraz OpenBSD (2.3), CSRG rozwiązane
- **1996:** OpenBSD (z NetBSD)

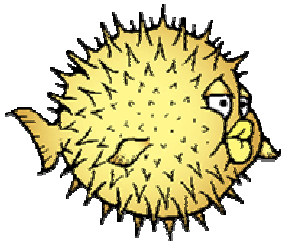


Dziś

- OpenBSD 4.0 FreeBSD 6.1 NetBSD 3.1



freeBSD®



OpenBSD





logo NetBSD





logo NetBSD



Japonia 1945



Plan prezentacji

1. Historia BSD
- 2. Licencja BSD a GNU/GPL**
3. Brak dystrybucji – wada czy zaleta ?



Licencja FreeBSD (fragment)

Redystrybucja i wykorzystanie w postaci źródłowej i formach binarnych, **wraz lub bez modyfikacji**, rozpowszechnianie **są dozwolone** przy spełnieniu następujących warunków:



Licencja FreeBSD (fragment)

Redystrybucja i wykorzystanie w postaci źródłowej i formach binarnych, wraz lub bez modyfikacji, rozpowszechnianie są dozwolone przy spełnieniu następujących warunków:

1. Redystrybucja kodu źródłowego musi zachować wyżej zamieszczoną notę o prawach autorskich, obecny wykaz warunków (...).
2. Redystrybucja w formie binarnej musi zawierać wyżej wymienioną notę o prawach autorskich, obecnym wykazie warunków (...).



Licencja GNU/GPL (fragmenty)

1. **Możesz kopiować i rozprowadzać** (...) wierne kopie kodu źródłowego Programu (...) pod warunkiem, że (...) podasz na każdej kopii właściwą informację o prawie autorskim [...]



Licencja GNU/GPL (fragmenty)

1. Możesz kopiować i rozprowadzać (...) wierne kopie kodu źródłowego Programu (...) pod warunkiem, że (...) podasz na każdej kopii właściwą informację o prawie autorskim [...]
2. **Możesz modyfikować** swoją kopię czy kopie Programu oraz dowolne jego części, tworząc przez to pracę opartą na Programie, jak również **kopiować i rozprowadzać takie modyfikacje** i pracę na warunkach podanych w pkt.1 powyżej - pod warunkiem [...]:



Licencja GNU/GPL (fragmenty)

1. Możesz kopiować i rozprowadzać (...) wierne kopie kodu źródłowego Programu (...) pod warunkiem, że (...) podasz na każdej kopii właściwą informację o prawie autorskim [...]
2. Możesz modyfikować swoją kopię czy kopie Programu oraz dowolne jego części, tworząc przez to pracę opartą na Programie, jak również kopiować i rozprowadzać takie modyfikacje i pracę na warunkach podanych w pkt.1 powyżej - pod warunkiem [...]:
 - Musisz doprowadzić do tego, aby **każda rozpowszechniana** lub publikowana przez ciebie **praca**, która w całości lub części zawiera Program, albo pochodzi od niego lub jego części, **była w całości i bezpłatnie licencjonowana** dla wszelkich stron trzecich **na warunkach niniejszej Licencji**.
 - **Musisz dołączyć** do niego odpowiadający mu, kompletny i możliwy do odczytania przez urządzenia cyfrowe **kod źródłowy**, który musi być rozpowszechniany na warunkach pkt.1 i 2 powyżej



Porównanie

FreeBSD:

GNU/GPL:



Porównanie

FreeBSD:

- bardzo liberalna

GNU/GPL:

- ograniczenie liberalna



Porównanie

FreeBSD:

- bardzo liberalna
- skupiona na prawach użytkownika

GNU/GPL:

- ograniczenie liberalna
- skupiona na otwartości źródeł



Porównanie

FreeBSD:

- bardzo liberalna
- skupiona na prawach użytkownika
- brak wymagań względem prac pochodnych

GNU/GPL:

- ograniczenie liberalna
- skupiona na otwartości źródeł
- „wirusowa” – wymaga tej samej licencji dla prac pochodnych



Porównanie

FreeBSD:

- bardzo liberalna
- skupiona na prawach użytkownika
- brak wymagań względem prac pochodnych
- pozwala pobierać dowolne opłaty

GNU/GPL:

- ograniczenie liberalna
- skupiona na otwartości źródeł
- „wirusowa” – wymaga tej samej licencji dla prac pochodnych
- pozwala pobierać dowolne opłaty (np. komercyjne dystrybucje Linuxa)



Plan prezentacji

1. Historia BSD
2. Licencja BSD a GNU/GPL
3. Brak dystrybucji – wada czy zaleta ?



Brak dystrybucji - wady



Brak dystrybucji - wady

- brak konkurujących dystrybucji (problem z dopasowaniem do potrzeb użytkowników)



Brak dystrybucji - wady

- brak konkurujących dystrybucji (problem z dopasowaniem do potrzeb użytkowników)
- **mniejsze możliwości personalizacji**



Brak dystrybucji - wady

- brak konkurujących dystrybucji (problem z dopasowaniem do potrzeb użytkowników)
- mniejsze możliwości personalizacji
- brak łatwych dystrybucji dla początkujących



Brak dystrybucji - wady

- brak konkurujących dystrybucji (problem z dopasowaniem do potrzeb użytkowników)
- mniejsze możliwości personalizacji
- brak łatwych dystrybucji dla początkujących
- skłonność do pozostawania w tyle pod względem obsługi najnowszego sprzętu



Brak dystrybucji - zalety

- centralizacja wytwarzania oprogramowania



Brak dystrybucji - zalety

- centralizacja wytwarzania oprogramowania
 - staranne, długofalowe projektowanie (szybkość i stabilność)



Brak dystrybucji - zalety

- centralizacja wytwarzania oprogramowania
 - staranne, długofalowe projektowanie (szybkość i stabilność)
 - uporządkowana, systematyczna dokumentacja



Brak dystrybucji - zalety

- centralizacja wytwarzania oprogramowania
 - staranne, długofalowe projektowanie (szybkość i stabilność)
 - uporządkowana, systematyczna dokumentacja
 - zawiera tylko najważniejsze moduły (optymalizacja, brak przeładowania)



Brak dystrybucji - zalety

- centralizacja wytwarzania oprogramowania
 - staranne, długofalowe projektowanie (szybkość i stabilność)
 - uporządkowana, systematyczna dokumentacja
 - zawiera tylko najważniejsze moduły (optymalizacja, brak przeładowania)
- mała szansa na „zawieszenie projektu” (Aurox)



Brak dystrybucji - zalety

- centralizacja wytwarzania oprogramowania
 - staranne, długofalowe projektowanie (szybkość i stabilność)
 - uporządkowana, systematyczna dokumentacja
 - zawiera tylko najważniejsze moduły (optymalizacja, brak przeładowania)
- mała szansa na „zawieszenie projektu” (Aurox)
- brak problemów niekompatybilności dystrybucji



Brak dystrybucji - zalety

- centralizacja wytwarzania oprogramowania
 - staranne, długofalowe projektowanie (szybkość i stabilność)
 - uporządkowana, systematyczna dokumentacja
 - zawiera tylko najważniejsze moduły (optymalizacja, brak przeładowania)
- mała szansa na „zawieszenie projektu” (Aurox)
- brak problemów niekompatybilności dystrybucji
- wysiłki i wsparcie społeczności skupione na jednym produkcie