

The Parrot System

formerly Parrot Security OS

19 grudnia 2020

Adrianna Saribekyan, Ewelina Kowal, Łukasz Siemieniec

The Faculty of Materials Engineering and Physics
Cracow University of Technology

- 1 Wstęp

- 2 Dlaczego Parrot?

- 3 Historia

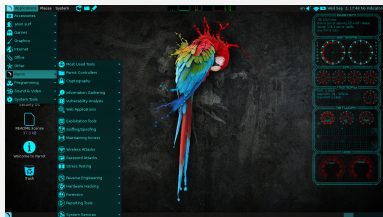
- 4 Cel Parrota

- 5 Bibliografia

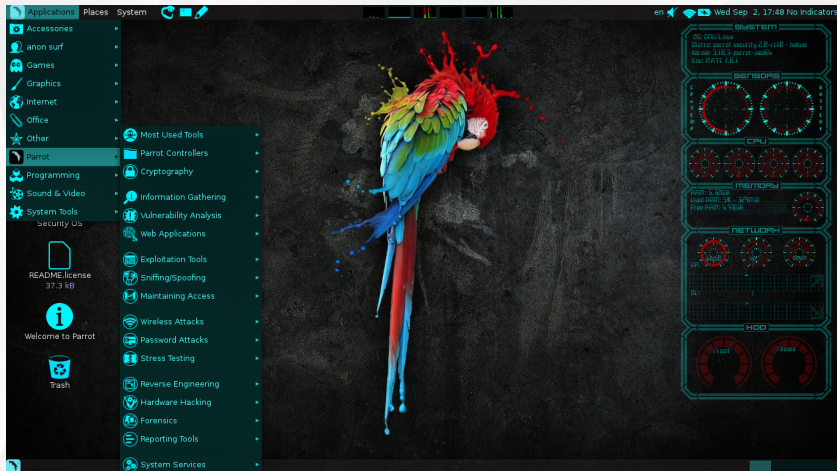
Parrot OS jest to darmowa dystrybucja Linux z otwartym kodem źródłowym który jest oparty na Debian Testing. Przeznaczony jest dla osób które zajmują się bezpieczeństwem w sieci, programistom oraz studentom.

Przedstawimy jak Parrot różni się od dystrybucji:

- Ubuntu
- Pentest
- Secure
- Forensics



Rysunek: Parrot 2.0 interfejs.



Rysunek: Parrot 2.0 interfejs.

	Parrot:
Ubuntu	daje dostęp do plików wewnętrznych
Pentest	zawiera programy codziennego użytku
Secure	aplikacje chronione przed sobą nawzajem
Forencis	nie otwiera automatycznie dysków.

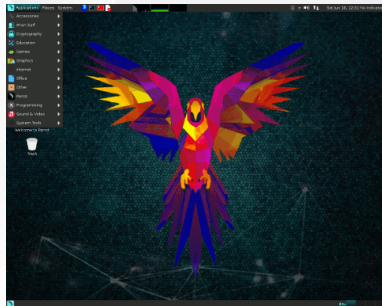
Tabela: Porównanie Parrot z innymi dystrybucjami.

Nazwa Parrot pochodzi od żartu prima aprilis, który zapowiadał hipotetyczny język o nazwie Parrot, który ujednoczyłby Pythona i Perla. Nazwa została później przyjęta przez ten projekt, który ma na celu obsługę m.in. Python'a i innych języków programowania. Kilka języków jest przenoszonych do pracy na maszynie wirtualnej.

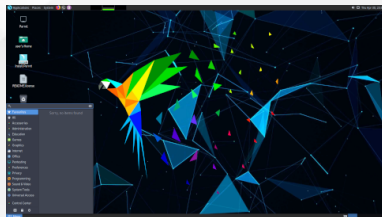
Języki

- Hostowanie języków klienta i umożliwienie współdziałania między nimi.
- Wykorzystuje obszernej literatury na temat optymalizacji kompilatora do generowania kodu bajtowego dla maszyny wirtualnej Parrot.
- Parser Grammar Engine, hybrydowy generator parsera.

Wygląd interfejsu



Rysunek: Starsza wersja: Parrot 3.0.



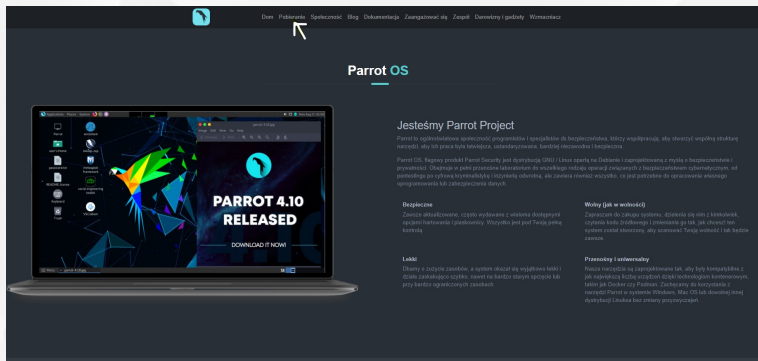
Rysunek: Najnowsza wersja: Parrot 4.9.

Języki statyczne i dynamiczne

Różne właściwości języków wpisywanych statycznie i dynamicznie były motywacją do zaprojektowania Parrot. Obecne popularne maszyny wirtualne, takie jak maszyna wirtualna Java i środowisko uruchomieniowe języka wspólnego dla platformy .NET, zostały zaprojektowane dla języków z typami statycznymi, podczas gdy języki docelowe dla programu Parrot są typowane dynamicznie.

Dlaczego Parrot?

- Bogate wsparcie dla kilku funkcji programowania funkcjonalnego
- Dynamiczna rozszerzalność obiektów metodami,
- Pozwala na stosowanie kontenerów polimorficznych i związanych z nimi opkodów
- Eliminuje potrzebę rozwiązywania problemów w poszczególnych językach.



Oficjalna strona dystrybucji.

parrotsec.org/download/

Home Download Community Blog Documentation Get Involved Team Donations & Gadgets Partners

Download ParrotOS 4.10

Signed Hashes Browse Archive Beta images

All Home Security Mate Kde Virtual

Parrot Home MATE ISO (default)

- > Direct Download
- > Select Server
- > Get Torrent

For Daily use No Security
Tools MATE desktop Size 1.8Gb

Parrot Home KDE ISO

- > Direct Download
- > Select Server
- > Get Torrent

For Daily use No Security
Tools KDE desktop Size 1.9Gb

Parrot Home OVA

- > Direct Download
- > Select Server
- > Get Torrent

For Virtual Machines Home
Edition MATE desktop Size 3Gb

Obsługiwane języki Od lipca 2013 r., aktywnie obsługiwane języki

to:

- C
- Java
- Java bytecode
- Raku
- Python
- Lua
- Perl 5
- Joy

Formy kodowania

Bytecode	binarny i natywnie interpretowany przez Parrot
PASM	jest językiem niskiego poziomu
PIR	zarządza wewnętrznymi konwencjami wywołań Parrot

- https://en.wikipedia.org/wiki/Parrot_virtual_machine
- <https://www.parrotlinux.org/docs/>
- <https://distrowatch.com/table.php?distribution=parrot>

Dziękujemy za uwagę