



Linuks Jest Niestabilnym Systemem

2011-2024 Wszelkie Prawa Zastrzeżone przez Jacka Marcina Jaworskiego czyli Energo Kodera Atlanta

autor:	Jacek Marcin Jaworski
pseudonim:	Energo Koder Atlant
pomocnicy autora:	BRAK
miejsce:	Pruszcz Gd., Polska
utworzono:	2011-12-17, sob.
wersja: 132 z dnia:	2026-07-19
program składu:	Libre Office Writer
sys. op.:	Windows 7
źródło:	energokod.gda.pl

Ten dok. w wer. PDF jest podpisany cyfrowo wolnym prog. GNU gpg dostępnym bezpłatnie ze s. www.gnupg.org/download. Instrukcja w j. pol. jak się posługiwać prog. GNU gpg w sys. Linuks z rodz. Debian/Ubuntu znajduje się w całkowicie bezpłatnym dok. PDF [Konf. i Zabezp. Sys. Op. z Rodz. Ubuntu](#) w roz. "Podpisywanie Dok. PDF".

Spis treści

Przedmowa Po 13 Latach Od Pierwszej Pub.....	1
Wstęp.....	1
O Autorze.....	1
Po Co Sys. Op.?......	2
Stabilność Sys. - Definicja.....	2
Aspekty Stabilności Systemu.....	2
Stabilność Linuksa.....	2
Niestabilność Patologiczna.....	3
Dowód.....	3
Nie Stabilność Przyczyną Porażki Na Rynku Komputerów Osobistych.....	3
Jak Programista Aplikacji Ma Sobie z Tym Radzić?.....	3
Komentarz.....	3
Historia Zmian.....	4
2011-12-17.....	4
2014-03-24.....	4
2020-09-16.....	4
2022-10-15.....	4
2024-12-17.....	4
2025-01-06.....	4
2026-07-18.....	4
Licencja.....	4

Przedmowa Po 13 Latach Od Pierwszej Pub.

Od 2011-12-17, sob. wiele się zmieniło na lepsze w świecie sys. op. Linuks. Przede wszystkim wprowadzono wydania LTS, które pozwalają spokojnie używać tych sys. przez 10 a nawet więcej lat. Jest to b. potrzebne wszystkim zawodowcom. Jednak jest problem z tymi wer. LTS, bo nie tylko wprowadza się w nich łaty wykrytych błędów, ale dodaje się do nich nowe wer. prog. (np. prog. gł. Linuks, przeglądarki Ognisty Lis i klienta pocztowego Grzmiący Ptak). Moim zdaniem jest to całkowicie nie dopuszczalne i sprzeczne z logiką. Po 2010 wyraźnie poprawiła się stabilność prog. dostarczanych z sys. Linuks.

Po mojej ucieczce z padaczkowych dystrybucji z okresu przed 2010r., z końcem 2017r. wróciłem do Linuksa i używam go co dziennie do dziś (piszę to w d. 2025-01-06, pon.).

Wstęp

Udowodnię tu że GNU Linuks nie jest stabilnym systemem (w brew propagandzie) i dla tego jest tak mało popularny na komputerach osobistych.

O Autorze

Od ur. w 1978 r. w Gd. żyję w Pruszczu Gd. na ul. Space-rowej. Od 2020-06-20 do 2021-06-19 mieszkam w Gdyni na ul. Komandorskiej (dokładnie po roku wróciłem do Pruszcza Gd.).

Mój życiorys zawodowy jest dostępny na mojej s. WWW w dok. [Najnowsze Podanie o Pracę.pdf](#). Natomiast pełen przebieg mojej kariery naukowej i zawodowej (łącznie ze sposobem wrobienia mnie w schizofrenię przez kol. z Tł. w Gd.) przedstawiam w [14. Raport Totalizacyjny: Nauka i Praca Programisty C++ w III Rzeczy \(pospolitej\).pdf](#). W tym roz. podaję jedynie skrócone info na mój temat:

Programuję komputery od lut. 1997 r. Od 1999 r. mam tytuł Technika Elektronika spec. Systemy Komputerowe. Zaliczyłem 3 lata wieczorowych i nielegalnych studiów inżynierskich na PG (jednak dyplomu inż. nie zrobiłem). Legalną maturę zdałem w 2016 r. Jednak po 2005 r. wprowadzono w całej Polsce zmiany w prog. studiów zaocznych (wymuszono na wszystkich kier. 7 sem. z j. ang.) powoduje to, że nie jestem w stanie ich zaliczyć.

W latach 1999-2007 (jednak bez ciągłości zatrudnienia) byłem programistą aplikacji w C++ dla sys. Windows.

W latach 2017-2022 (jednak bez ciągłości zatrudnienia) byłem programistą aplikacji w C++ na sys. Linuks i Android.

Jeśli chodzi o mój kierunek rozwoju, to chcę być jak najlepszym inżynierem i architektem sys. i to nie tylko komputerowych.

Moje zainteresowania zawodowe to przede wszystkim:

*Studia nad architekturą wzorcowych sys. komp.*Studia nad nowymi algorytmami (jednak nie SI)*Studia nad bezpieczeństwem współczesnych sys. komp.*Studia nad prywatnością użytkowników współczesnych sys. komp.*Podnoszenie jakości, efektywności i bezpieczeństwa w pracy przez realizację racjonalizatorskich projektów*

Programowanie w językach: Asembler, C i C++.

Wystrzegam się jak mogę „produktów” wielkich korporacji, które z niejawnych (zło jest niejawne) powodów zwykle wynajdują patologiczne wynalazki¹.

Jestem poszukiwaczem i kolekcjonerem "dobrych zasad życiowych" i "dobrych zasad inżynierskich". Dzięki tym związyłem, hasłowym zasadom często widzę sens podejmowania większego wysiłku by uzyskać obiektywnie dobry efekt zamiast stosowania półśrodków.

Zdrowe zasady pozwalają zostawiać za sobą działające rozwiązania zamiast partactw.

Jestem praktykującym zwolennikiem filozofii dobra i moralności, czyli Totalizmu. Od końca 2001r. mam stały dostęp do Internetu. Z filozofią Totalizmu, w wyd. prof. Jana Pajęka z NZ, po raz pierwszy zetknąłem się w Sieci Internet w I poł. 2002r., po wpisaniu w Google.pl hasła "ESP" - bez cudzołowu. Od razu b. się zafascynowałem filozofią Totalizmu i tak jest do d. dzisiejszego. Od 2020r. rozwijam jej popr. wer. w postaci [Ideologii Geniuszy-Mocarzy](#). Mój Totalizm różni się od Totalizmu prof. Jana Pajęka z NZ tym, że ja uważam że Totalizm składa się z dobra i z moralności, a nie z samej moralności. Wynika to z faktu, że sama moralność bez dobra prowadzi do filozofii moralności i zła czyli w istocie do filozofii nazistowskiej.

W latach 2004-2013 5x siedziałem w psychiatryku. Sądy i lekarze byli i są jednomyślni że jestem b. chory psychicznie, na przekór braku jakichkolwiek dowodów mojego rzekomego szaleństwa. Za "dowody" mojego szaleństwa lekarze uznają to, że otaczających mnie ludzi mam za "wrogich szatańskich pasożytów" i że po wzmocnionych dawkach leków psychotropowych (bo w szpitalach mogą dawać mocniejsze dawki) spałem całymi dniami - nazwali to "zespołem katatonicznym - paranoidalnym w przebiegu schizofrenii".

W d. 2025-06-17, wto. sąd rej. w Gd. skazał mnie na 6. odsiadkę w psychiatryku bez udowodnienia mi żadnych win oraz wcale nie odnosząc się do mojej obronnej argumentacji

- 1 Jak wiecie język **C** i sys. **Uniks** stworzyło 2 ludzi: **Dennis Ritchie** i **Ken Thompson**. Dlatego jestem na 100% pewien, że gdyby ich projekt był prowadzony z rozmachem w stylu wielkich korporacji, to: a) Język **C** nigdy by nie powstał, b) **Uniks** nigdy by nie powstał, c) powstały by potwory podobne do **Ada**, **Jawa**, **C#**, **Windows** i **Android**.

Jak zauważył prof. Jan Pajęk z NZ: W organizacjach pasożytniczych wszystkie złe pomysły pojawiają się od góry, a wszystkie postępowe koncepcje powstają od dołu.

tacji jasno wykazującej że to oskarżenie (personel domowy i medyczny) "ma coś z dekle" a nie ja. Sąd okręgowy w Gd., na posiedzeniu niejawnym, w d. 2025-10-27, pon. odrzucił moją apelację bez jej rozpatrzenia i podtrzymał wyrok sądu rej. Co dziwne tego wyroku na razie nikt nie wyegzekwował (piszę to w d. 2026-06-26, pią.). Przejawy mojej dyskryminacji oraz odrzuconą apelację do sądu okręgowego przytaczam w: [2025-10-06 Do Amnesty International Polska - Prośba o Pomoc Dla Prześladowanego Jacka Marcina Jaworskiego.pdf](#).

Od 2024-12-20, pią. prowadzę moją domową s. WWW: [energokod.pl](#). Od 2025-11-07, pią. jest ona dostępna pod nowym adresem [energokod.gda.pl](#). Na tej s. WWW pub. kilkadziesiąt dok. PDF o tematyce totaliztorskiej, Mini Netykietę i moje dane kontaktowe. Nowości na mojej domowej s. WWW można śledzić dzięki plikowi/kanałowi [energokod.gda.pl/rss.xml](#).

Po Co Sys. Op.?

System operacyjny miał przykrywać różnorodność współczesnych komputerów oraz (co równie istotne) różnorodność przyszłych komputerów przed programami użytkowymi.

Te programy miały być "bardziej trwałe" albo "bardziej żywotne" dzięki systemom operacyjnym.

Z czasem przyjęło się, że system operacyjny ma również dostarczać podstawowe aplikacje takie jak przeglądarka, program pocztowy czy odtwarzacz multimedialny. Ale te domyślne programy to tylko przysłowiowy "kwiatek do kożucha".

Stabilność Sys. - Definicja

Stabilność systemu: Jest to stopień w jakim system zachowuje swoją odpowiedzialność w czasie.

Aspekty Stabilności Systemu

W odniesieniu do systemów operacyjnych mamy do czynienia z dwoma aspektami stabilności systemu:

- 1 Stabilność czasu wykonania;
- 2 Stabilność udostępnianych funkcjonalności (trwałość interfejsów).

Stabilność Linuksa

W odniesieniu do stabilności czasu wykonania (ad .1) z jądrem Linuksa jest całkiem dobrze (b. rzadko się wie-sza). Natomiast ze stabilnością aplikacji użytkowych jest już dużo gorzej. Jednak dzięki pracy firmy Canonical systemy Ubuntu i Kubuntu dają się używać.

Więc w czym problem?

Problem jest ze stabilnością udostępnianych funkcjonalności przez system Linuks (ad 2). W świecie wolnego oprogramowania normą jest częsta zmiana interfejsów programistycznych.

Jakieś pół roku temu (w 2011r.) czytałem, że w jądrze Linuks pozmieniano obsługę USB i temu programiście co o tym pisał włosy dęba stały bo nie był w stanie uzyskać takiego zachowania jego aplikacji jak wcześniej. Wtedy w desperacji przeniósł swoje aplikacje na Free BSD.

Od początku brak jest interfejsów binarnych w kernelu Linuksa i w innych kluczowych pod systemach (np. w Kubuntu 11.04 nie można zainstalować sterownika nvidia-96 z powodu innego ABI w XWindows).

Ja odnoszę wrażenie, że twórcy aplikacji dla Linuksa chcą jak najszybciej tworzyć efektowne i nowoczesne aplikacje. Ciągłe przewalają kody swoich programów by dostosować je do znowu pozmienianych bibliotek. I choć niby gonią za nowościami to jednak ciągle siedzą w starych narzędziach typu Autotools czy Emacs nie wspominając już o języku C. Zaś bardzo potrzebne programy takie jak KDevelop i Umbrello latami nie mogły uzyskać akceptowalnej stabilności (i pewnie tak jest do dziś).

Niestabilność Patologiczna

Niestabilność GNU Linuks jest systemowa, czyli patologiczna. Bo wygląda na to, że Linus Torvalds nie rozumie jak krytyczne dla rozwoju systemu są stabilne (w sensie trwałe i dostępne przez długi czas) podstawowe interfejsy systemu. Niestety dla Linusa i dla wszystkich buntowników przeciw dominacji M\$ częścią stabilnego systemu są interfejsy binarne (ABI). Zaś obłąkańcze mamrotanie że tylko programy z dostępnym kodem źródłowym są akceptowalne to wpuszczanie całego ruchu użytkowników GNU Linuks w kanał. M\$ Windows to kanał, Linuks mógłby być alternatywą, ale przez jego nie stabilność to też kanał nawet gorszy, bo czyni ogromny wysiłek twórczy bez sensownym. Nic nie jest trwałe w świecie GNU Linuks wprowadza się nie kompatybilne zmiany w jądrze, w podsystemach systemu operacyjnego (XWindows) i szybko wprowadza się nowe biblioteki porzucając stare (np. Qt 3), narzędzia linii komend GNU też doświadczają niekompatybilnych zmian (sam się przekonałem że po roku 2004 zmieniło się wyjście programu wc - ciekawe z jakiego powodu??).

Żadna poważna firma nie może się oprzeć na distro do którego poprawki wychodzą tylko przez 2 lata. Poważne systemy to takie których działanie jest stabilne i stabilne są funkcje jakie udostępniają one aplikacjom. Kiedyś w szkole uczyłem się, że elektronika komputerowa jest projektowana do pracy przez 10 lat. Tak więc logicznym jest, że system operacyjny musi udostępniać swoje funkcjonalności przez dłuższy czas niż te 10 lat działania sprzętu, bo musi być czas na przygotowanie oprogramowania do pracy pod nowymi systemami. Ile to może być czasu? 2 lata to raczej mało, więc jakieś 3-4 lata zapasu musi być na to dostosowanie. Zatem interfejsy systemu operacyjnego muszą być stabilne i dostępne przez 13-14 lat minimum - żeby mówić o poważnym systemie na którym

można opierać własną przyszłość (prowadzić firmę, czy zarabiać na usługach komputerowych).

Dowód

Istnieją takie systemy operacyjne które zachowują wysoką stabilność na przestrzeni kilkunastu i więcej lat. Systemy rodziny Windows NT dostarczają tych samych funkcji od połowy lat 90 XX w. czyli już jakieś 15 lat i podejrzewam, że oprogramowanie pisane na Windows NT 3.1 bez jakichkolwiek zmian będzie śmigać u mnie na Windows 7 32bit. Trochę dziwne jest, ale Microsoft zapowiedział, że wsparcie dla Windows XP będzie realizował do 2014 roku, czyli będzie to właśnie postulowane przeze mnie 13 lat! Tak więc ze względu na zadowalającą stabilność czasu wykonania jak i na bardzo wysoką stabilność (trwałość) udostępnianych interfejsów Microsoft jest bardzo poważnym partnerem biznesowym. No ale może ktoś mi zarzuci że wyjechałem z M\$ jak jakiś urzędnik. OK! Więc przywołam inną firmę. IBM swój DOS/360 dostarczał od 1966 roku do 2005! Czyli niemal 40 lat! Zatem istnieje zapotrzebowanie na stabilność udostępnianych funkcji (trwałość interfejsów) i to zapotrzebowanie jest równie wielkie jak stabilność czasu wykonania.

Nie Stabilność Przyczyną Porażki Na Rynku Komputerów Osobistych

Gdyby systemy Linuks były stabilne choćby na przestrzeni ostatnich 15 lat, to prawie na pewno komercha by zaczęła dostarczać znaczące ilości aplikacji - szczególnie administracyjnych, biznesowych, inżynierskich i gier. Bo po prostu to by się coraz bardziej opłacało (bo przecież teoria jest taka, że kapitaliści są gotowi na wszystko byle cię koszty). Zwykli ludzie widząc rozsądek w pracy z tym systemem też by go używali. Natomiast teraz sytuacje mamy taką, że za te 120\$ jakie trzeba dać Microsoftowi otrzymujemy na prawdę to co potrzeba by dało się pracować, uczyć i bawić.

Jak Programista Aplikacji Ma Sobie z Tym Radzić?

Dbaj po prostu o to by twoje programy były przenośne między systemami operacyjnymi. W najważniejszych j. programowania da się tak programować by kod był przenośny.

Z zasady nieprzenośne są wynalazki M\$ dlatego należy się trzymać od nich z daleka.

Komentarz

Nie jestem wielbiciele M\$ - ponad 10 lat walczyłem z GNU Linuksem zanim stwierdziłem, że są z nim większe problemy niż mogło by się wydawać. Teraz zdiagnozowa-

łem ten problem i było to dla mnie sporym zaskoczeniem, że GNU Linuks jest nie stabilny, bo wcześniej bezmyślnie wierzyłem w legendę, że jest stabilny.

Jestem zadowolony z tego przełomu intelektualnego jakiego udało mi się dokonać.

Przykre jest to, że wielu ambitnych ludzi często z wielkim talentem i umiejętnościami traci czas na coś co z definicji nie może być dać efektu na tyle trwałego, by można było powiedzieć, że ich praca jest sensowna. Tym nie mniej swoją skromną pracą (nie płatną) wspieram również użytkowników systemu Linuks.

Historia Zmian

2011-12-17

Opublikowanie pierwszej wersji tego dokumentu na grupie dyskusyjnej pl.comp.os.linux.

2014-03-24

Zredagowanie tego dokumentu w formie strony internetowej i zamieszczenie jej na serwerze szyk.jcom.pl.

2020-09-16

Dostosowanie do publikacji w formie monografii (poprawki składu, poprawki stylistyczne). Dodany akapit „Jak programista aplikacji ma sobie z tym radzić?”.

2022-10-15

Spolszczyłem terminy. Poprawiłem literówki.

2024-12-17

Poprawiłem literówki.

2025-01-06

Dodałem roz. Przedmowa po 13 latach od pierwszej pub.

2026-07-18

Dodałem roz. O Autorze.

Licencja

Jest to licencja dotycząca tego dokumentu. Pliki wskazane przez linki mogą być publikowane na innych licencjach. Zasady licencji:

1. **Zezwolenie na kopiowanie** Zezwala się na niekomercyjne kopiowanie tego dokumentu;
2. **Zezwolenie na udostępnianie** Ten dokument można udostępniać (jednak bezpłatnie);

3. **Zabronione modyfikowanie** Tego dokumentu nie można modyfikować ani skracać ani dodawać cokolwiek.
4. **Ograniczenia licencji nie dotyczą autora.**