

15. Raport Totalizacyjny: Średniowiecze Po Zniszczeniu AmigaOS i Plan9

1 Metadane

	autor:	Jacek Marcin Jaworski
	pseudonim:	Energó Koder Atlant
	pomocnicy autora:	BRAK
	miejsce:	Pruszcz Gd.
	utworzono:	2021-11-15 (org. przed ustaleniem standardu Rap. Tot.)
	wersja: 256 z dnia:	2025-05-10
	program składu:	Libre Office Writer
	sys. op.:	Kubuntu

Spis treści

1	Metadane.....	1
2	Wstęp.....	2
2.1	Skróty.....	2
2.2	Teza.....	2
2.3	Streszczenie.....	2
3	Mój oryginalny wkład.....	3
4	Metoda badawcza.....	3
5	Fakty.....	3
5.1	Przełomowe technologie jakie obecnie są niedostępne.....	3
5.1.1	Skrypty w j. AREXX z sys. op. AmigaOS 2.0 (i nowsze).....	3
5.1.2	Praca na klastrze w sieci z komp. z sys. op. Plan9.....	3
5.2	Jak oprogramowaniem psuje się ludzióm umysły.....	4
5.3	Jak popsutym oprogramowaniem uniemożliwia się wydajną pracę.....	4
5.4	Dzisiejsze oprogramowanie nie jest podatne na rozszerzanie.....	5
5.5	Dzisiejsze oprogramowanie jest niekompletne.....	5
6	Moralna ocena faktów przytoczonych w raporcie.....	5
6.1	Z czego wynikają kryteria oceny.....	5
6.2	Jakie Wsk. Moralnej Popr. "są na tak" a jakie "są na nie" i dlaczego?.....	5
6.3	Analiza znanych faktów w świetle jednego, wybranego Wsk. Moralnej Popr.....	10
7	Podsumowanie.....	10
7.1	Rewelacje, zalety, wady i partactwa w udostępnianym oprogramowaniu.....	10
7.1.1	Rewelacje.....	10
7.1.2	Zalety.....	10
7.1.3	Wady.....	11
7.1.4	Partactwa.....	11
7.2	Wnioski.....	11
7.3	Czy problem jest organizacyjny czy jednostkowy?.....	11
7.4	Czy problem jest trwały czy czasowy?.....	11
7.5	Przewidywania na przyszłość.....	11
7.6	Zalecenia na teraz.....	11
7.7	Zalecenia na przyszłość.....	11
8	Licencja.....	11
9	Bibliografia.....	12

2 Wstęp

Powodem napisania tego raportu jest stagnacja w dziedzinie sys. op. która nastąpiła po bankructwie amerykańskiej firmy [commodore-s-w-wiki] w kwi. 1994r.

2.1 Skróty

API	interfejs programistyczny (z j. ang. Application Programming Interface)
d.	dzień
dok.	dokument
f.	funkcja
GUI	interfejs użytkownika (z j. ang. Graphic User Interface)
jw.	Jak wyżej
kol.	kolejny
org.	oryginalny
pam.	pamięć
PC	komputer osobisty (w j. ang. Personal Computer)
poz.	pozycja
prog. gł.	program główny (w j. ang. kernel)
pyt.	pytanie
s.	strona
sys. op.	system operacyjny
tel.	telefon
ust. dom.	ustawienia domyślne
wer.	wersja
woj.	województwo
Wsk. Moralnej Popr.	Wskaźniki Moralnej Poprawności
SZAP	Stany Zjednoczone Ameryki Północnej

Skrótów 4literowych i dłuższych nie tłumaczę, bo uważam je za oczywiste.

2.2 Teza

W treści wykażę, że znane są takie rozw. informatyczne które nadają pracy z komputerem zupełnie inny, wyższy poziom techniki niż ma to miejsce obecnie.

Druga sprawa do wykazania to fakt, że obecne, niewygodne bo popsute GUI i API powodują psucie umysłu.

Trzecia sprawa do wykazania to fakt, że obecnie dostępne prog. nie są tworzone z myślą o ich łatwym rozszerzaniu.

Czwarta sprawa do wykazania to fakt, że udostępniane z sys. Linuks prog. mają niekompletną funkcjonalność. Jednak wykazanie, że cywilom udostępnia się porzucone, nie trafione sys. i prog. jest obecnie poza moimi możliwościami.

Piąta sprawa do wykazania to fakt, że przez psucie oprogramowania uniemożliwia się wydajną pracę z komputerem.

2.3 Streszczenie

jw. w postawionych tezach.

3 Mój oryginalny wkład

W latach 1993-1995 byłem użytkownikiem komputera Commodore Amiga 600. W latach 90. XXw. Byłem czytelnikiem mieś. Magazyn Amiga i Chip PL. Od gru. 1996r. Jestem użytkownikiem komp. PC. Od lut. 1997r. prog. komp. Od 1999r. pracowałem w kilku firmach woj. Pomorskiego (w Polsce) jako programista prog. na PC z sys. Widnows (pocz. 95 i 98, a później NT i jego wer. rozwojowe). W Latach 2017-2022 pracowałem jako programista prog. na sys. wbudowane Linuks i Android w różnych firmach woj. Pomorskiego. Od 2021-05 do 2022-08 byłem użytkownikiem komputerka Vampir 4+ Standalone (kompatybilny z Amiga 1200) stworzony w niemieckim oddziale firmy [ibm-s-w-wiki]. Obecnie w sty. 2025r. nadal programuję i rozwijam swoje umiejętności i metody programowania.

4 Metoda badawcza

Analiza opiera się na moim wieloletnim doświadczeniu programisty i użytkownika komputerów. Najistotniejsze wspomnienia staram się popierać cytatami ze s. WWW (w tym z [s-wiki]).

5 Fakty

5.1 Przełomowe technologie jakie obecnie są niedostępne

Podam 2 przykłady przełomowych technologii jakie radykalnie zwiększają wydajność pracy z komputerem: AREXX i lokalny klaster.

5.1.1 Skrypty w j. AREXX z sys. op. AmigaOS 2.0 (i nowsze)

Amiga OS został zaprezentowany wraz z komputerem Amiga w 1985r. W przeciwieństwie do sys. op. Uniks nie miał on jednej wspólnej przestrzeni dyskowej tylko każdy nośnik był widoczny w sys. jako osobny napęd (tak jak w DOS i Windows). Najważniejszą wadą sys. op. AmigaOS jest brak ochrony przed modyfikacją pam. prog. gł. i pam. innych procesów działających w sys.

W roku 1990[amigaos-s-w-wiki] miała miejsce premiera AmigaOS 2.0 z j. skryptowym AREXX wymienionym na Workbench z IBM [amigaos-influence-s-w-wiki]. Język AREXX ma b. ciekawą możliwość: on umożliwia sterowanie prog. z GUI (okienkowych) za pomocą skryptów. To umożliwia pisanie skryptów które automatyzują wieloetapowe procesy z użyciem wielu prog. z interfejsem AREXX (te prog. muszą być specjalnie do tego przystosowane). Obecnie w sys. Linuks nie ma takich możliwości, a do pisania skryptów konieczne są prog. konsolowe (albo emulacja klikania która jest zawodna przy zmianie poz. lub roz. okna).

5.1.2 Praca na klastrze w sieci z komp. z sys. op. Plan9

[plan9-s-w-wiki] miał premierę w 1992r. Jest to następca sys. op. Uniks z 1970r. Sys. Plan9 tworzyli weterani, którzy wcześniej stworzyli sys. Uniks. Co było powodem powstania sys. op. Plan9? Powody były co najmniej dwa:

1. Chęć stworzenia sys. op. w którym wszystkie urządzenia były by plikiem. Należy tu powiedzieć, że w org. sys. Uniks (i podobnie we współczesnym sys. Linuks) karty sieciowe nie są plikami ani plikami nie są procesory (ani ich rdzenie), uruchomione prog. też nie są mogą być kontrolowane za pomocą f. sys. plików (w sys. Uniks brak jest dostępu do procesów przez pliki, w sys. Linuks jest to dostępne, ale w obu sys. nie ma dostępu do okien z poziomu f. sys. plików);
2. Chęć wykorzystania sieci Internet do współdzielenia plików (w tym urządzeń).

OBA TE CELE PROGRAMIŚCI SYS. PLAN9 OSIĄGŃĘLI W 100%! TAK TO NIE POMYŁKA: PROJEKT PLAN9 ZAKOŃCZYŁ SIĘ PEŁNYM SUKCESEM! Ten sys. pozwala na osiągnięcie w prosty sposób rzeczy obecnie nie możliwych lub możliwych tylko częściowo dla zawodowców po spec. studiach. Co np.? Np. można wyeliminować routery z sieci i tylko podłączać jeden PC do sieci globalnej a pozostałym kompom w sieci lokalnej udostępniać jego kartę sieciową. Podobnie nie potrzeba żadnych VPN, bo można w sieci Internet po prostu udostępniać swoją kartę sieciową.

ALE NIE TO JEST NAJLEPSZE W SYS. PLAN9! NAJLEPSZE JEST TO, ŻE MOŻNA WSPÓŁDZIELIĆ PROCESORY. DZIĘKI TEMU MOŻNA PRACOWAĆ NA WŁASNYCH KLASTRACH OBLICZENIOWYCH. Wystarczy podmontować do własnego sys. procesory kolegów z sieci lokalnej (lub procesory kolegów z globalnej sieci Internet). Dzięki temu można wykorzystać ich rdzenie w np. w celu szybszej kompilacji, albo do kompresji/dekompresji dużych ilości plików (np. do robienia paczek z programami albo przy tworzeniu kopi zapasowej), albo do renderowania scen 3D, albo do montażu filmów.

NA PRZEKÓR SUKCESOWI PROJ. PLAN9, OBECNIE (W 2025R.) NIE MA MOŻLIWOŚCI DO PROSTEJ PRACY NA WŁASNYCH KLASTRACH OBLICZENIOWYCH. WYGLĄDA NA TO, ŻE TĄ TECHNOLOGIĘ UTAJNIONO. A WIADOMO Z FILMÓW EMITOWANYCH W TV, ŻE TAJNYCH SYS. OP. W SZAP JEST DUŻO WIĘCEJ NIŻ TYCH KTÓRE DAJĄ CYWILOM.

5.2 Jak oprogramowaniem psuje się ludziom umysły

Ktoś wredny zauważył, że nawet nadmierne klikanie szkodzi na umysł. To dlatego przy wchodzeniu na każdą s. WWW każą nam klikać jakieś debilne zgody. Ale to zjawisko jest szersze i opiera się na wciskaniu prog. które łamią zasadę „zero-conf”, która oznacza, że prog. jest b. wygodny w użyciu w dom. konf. Oprócz przeglądarek wiele potrzebnych prog. wymusza zbędne kliknięcia. Po to by nadmierne klikać są robione wszelkie ekrany powitalne i podpowiedzi na starcie prog.

Oczywiście na tym nie koniec, bo psuje się też interfejsy API. Szczytem perfidii jest COM wynaleziony w SZAPowskiej firmie M\$. Ktoś szatański w M\$ zauważył, że COM jest tak skomplikowany, że nic się nie da zrobić bez wsparcia pomocy z M\$. Jak się o tym kapnęli, to wszystkie API w Windows przepisali na COM.

5.3 Jak popsutym oprogramowaniem uniemożliwia się wydajną pracę

Programowanie z tymi COM wyglądało (w 2 różnych firmach) tak, że rano odbierało się wskazówki z Ameryki, do południa się szarpało z tymi wskazówkami i w południe pisało się kol. pyt. do Ameryki. Oczywiście takie podejście uniemożliwia efektywną pracę. Dlatego jak jest taka możliwość, to należy dać stówę za coś normalnego i unikać jak ognia wszystkiego co jest z M\$, bo z M\$ nigdy nic normalnego nie było i nie będzie.

Wiele darmowych prog. dostarczanych z sys. op. Linuks jest celowo źle zrobionych, żeby ludziom mózgi lasowały. Np. Thunderbird pobiera sekwencyjnie wiad. z grup dyskusyjnych mimo że procesory są wielordzeniowe, a sieci wielokrotnie szybsze niż dawniej. Czyli wcale nie korzysta z pracy równoległej ani z szerokopasmowego dostępu do sieci Internet. Podobnie sekwencyjnie działa newsboat.

Inny problem dotyczy wszelkiego rodz. edytorów i przeglądarek PDF. Monitory są szerokie, a te prog. wyświetlają tylko 1s. naraz (z wielkimi marginesami po bokach).

Niektóre prog. (np. Qt Creator) co jakiś czas same robią coś złośliwego (np. przełączają tryb widoku proj., same uaktywniają jakieś mało istotne w danym momencie kontrolki, albo często resetują listę śledzonych zmiennych do ust. dom.) - to zmusza do zbędnych kliknięć, co denerwuje i oglupia. Ten Qt Creator nie umożliwia uruchomienia skryptu czyszczącego (albo kompilującego wtyczki) przed uruchomieniem prog. ze śledzeniem (w j. ang. debug). Co też zmusza do zbędnych kliknięć bo trzeba się przełączać na konsolę i ręcznie uruchamiać te skrypty.

Do tego wszystkiego obecnie widać globalną znowę, żeby uniemożliwiać programowania i zmuszać do pisania skryptów. Już teraz na sprytne tel. robi się wyłącznie apki skryptowe, natomiast na PC zamiast normalnych programów oferuje się apki dostępne przez przeglądarkę. Kluczowej bibl. Qt od lat już się wcale nie rozwija (zamiast niej qt.io rozwija skryptowy QML – potwór podobny do Java Skrypt, ale od niego jeszcze gorszy).

MOGĘ DODAĆ, ŻE JA JAKO PROGRAMISTA NORMALNYCH PROGRAMÓW NA PC OD 2022R. WCALE NIE MOGĘ ZNALEŹĆ PRACY W POLSCE. TAKIE ROZMIARY MA TA ZMOWA I PROPAGOWANIE SZALEŃSTWA SKRYPTOWEGO MAJĄCEGO ZASTĄPIĆ NORMALNE PROGRAMY.

5.4 Dzisiejsze oprogramowanie nie jest podatne na rozszerzanie

Taki Qt Creator ma wtyczki, ale nie ma żadnego dokumentu jak takie wtyczki tworzyć. Gdy się do nich pisze w tej sprawie chamsko każą się odwalić słowami „Zobacz sobie jak jest zrobiona jakaś gotowa wtyczka.”. Podobnie jest z Kdevelop (nawet gadka jest ta sama) i M\$ Visual Studio (to wiem od Qt Group, bo zawsze informują, że wraz z nową wer. VS przestała działać wtyczka do Qt, a po jakimś czasie ogłaszają tryumfalnie że wtyczka znowu działa). Ponoć powodem tego jest to, że interfejsy tych wtyczek szybko się zmieniają. I tak jest od dziesięcioleci. Dlatego mimo milionów użytkowników tych prog. niezależnych wtyczek jest b. mało. To tylko przykład, ale b. obrazowy.

Osobliwym zabezpieczeniem przed rozszerzaniem oprogramowania jest blokowanie w prog. konsolowych możliwości użycia ich w skryptach (robią to przez wykrywanie rodz. konsoli na której działają). Takie zabezpieczenie na 100% mają prog. tcpdump, pdfgrep i atrm. Tu dostęp do kodu źródłowego niewiele zmienia – tak jest on koszmarnie zagmatwany.

5.5 Dzisiejsze oprogramowanie jest niekompletne

Weźmy pod uwagę prosty przykład programów konsolowych sed i grep. Sed jest tekstowym edytorem strumieniowym, a grep jest wyszukiwarką tekstów. Oba mają b. dziwną wadę: mogą działać jedynie na pojedynczych liniach. Jest to sztuczne ograniczenie, bo czasem trzeba wyszukiwać i zamieniać napisy wieloliniowe. Wymaga to wczytywania całego pliku przed operacją wyszukiwania lub zamiany. Nie jest to nic niezwykłego, bo tak działa np. program konsolowy sort.

6 Moralna ocena faktów przytoczonych w raporcie

6.1 Z czego wynikają kryteria oceny

cytat z [Ideologia Geniuszy-Mocarzy]:

„ Badania prof. Pająka po godzinach: W toku tych badań naukowych prof. Jana Pająk ustalił jaka jest "filozofia Boga" czyli jak się ocenia ludzkie działanie. Te kryteria to podane w [totalizm 8p, tom 1, s. 31] "Wskaźniki Moralnej Poprawności". Prof. Pająk podał 8 pierwszych wsk. (w tym roz. są one podane jako pierwsze). Mi do tej pory udało się wskazać kolejnych 8. Na 2 wsk. (Droga i wdrożenia oraz Sukcesy naukowe i kompetencje) naprowadził mnie anonimowy Szaraczek. Szaraczek miał też udział w odkryciu wsk. Kultura Osobista, ale w nieco zawężonej postaci zwanej Samokontrola (która jest miarą jakości Kultury Osobistej).

Aby wykazać, że ktoś jest moralny Wsk. Moralnej Popr. muszą stwierdzać TAK!!! lub BRAK i nigdy NIE! Gdy choć jeden wsk. powie NIE! to znaczy, że ktoś lub coś jest niemoralne.”

6.2 Jakie Wsk. Moralnej Popr. "są na tak" a jakie "są na nie" i dlaczego?

Wsk. Moralnej Popr.	Skrótowy opis Wsk. Moralnej Popr.	Skrótowa analiza zagadnienia	Ocena
---------------------	-----------------------------------	------------------------------	-------

1. Sprawiedliwość i karma uczuciowa!!!	Moralne jest przestrzeganie zasady „jaka praca taka płaca”. Konieczne jest doświadczenie takich samych uczuć jakie się wzbudziło u innych. Moralne jest wzbudzanie u innych takich uczuć jakie sam chciał bym doświadczyć. Nie moralne jest robienie innym tego co mi jest niemiłe.	Psucie tego za co się bierze forszę jest skrajnie nie moralne (tak robi M\$ wymuszając używanie bibl. COM). Tworzenie gratisowego, wadliwego oprogramowania rozdawanego z sys. Linuks powoduje koszmarną karmę.	NIE!
2. Pole moralne!!!	Moralne jest podejmowanie wysiłku intelektualnego (wykazywanie się wysoką inteligencją). Niemoralne jest głupie postępowanie.	Tworzenie szatańskiego, bo wadliwego i psującego umysł oprogramowania jest chorobą moralną i umysłową decydentów i słuchających ich programistów. Jest to ujemna inteligencja (czyli szatańska).	NIE!
3. Totaliztyczne dobre uczynki i totaliztyczne grzechy!!!	To samo co wsk. Praca Moralna z uproszczeniem, że żyjemy w idealnym Świecie.	Programowanie to nie dobry ani zły uczynek tylko praca moralna bądź niemoralna.	BRAK
4. Praca Moralna!!!	Moralna jest jest dobrowolna praca, najbardziej pod górę pola moralnego i generująca dobrą karmę uczuciową. Nie moralna jest praca niewolnicza – jej wyniki są przeklęte.	Tworzenie gratisowego, wadliwego oprogramowania to krecia robota a nie praca moralna.	NIE!
5. Uczucia!!!	Uczucia moralne: 1. Przyjemne uczucia umysłowe są moralne; 2. Przyjemności fizyczne są niemoralne; 3. Dobrowolne pokonywanie przeciwnych uczuć jest moralne; Krzywdzenie istot postępujących moralnie jest niemoralne.	Tworzenie gratisowego, wadliwego oprogramowania to to czysta nienawiść!	NIE!

6. Motywacje!!!	Moralne jest podejmowanie wysiłku i przełamywanie lenistwa. Niemoralne jest uleganie lenistwu. Niemoralne jest uleganie korupcji w zamian za rezygnację z budowy własnej potęgi.	Tworzenie gratisowego, wadliwego oprogramowania to szatańskie motywacje!	NIE!
7. Odpowiedzialność!!!	Moralne jest branie odpowiedzialności za siebie i za innych. Niemoralne jest unikanie odpowiedzialności za swoje czyny.	Trudno powiedzieć za co chcą brać odpowiedzialność decydenci i programiści tworzący popsute prog.	BRAK
8. Sumienie!!!	Moralne jest słuchanie sumienia, bo zna ono wszystkie prawa moralne i wydaje nieomyłne sądy na temat każdego zrealizowanego działania. Niemoralne jest ignorowanie czyli zagłuszanie sumienia.	Tworzenie gratisowego, wadliwego oprogramowania jest sprzeczne z sumieniem!	NIE!
9. Uczciwość!!!	Moralna jest uczciwość. Moralny jest podstęp czyli iluzja przyszłości. Moralny jest psikus czyli żart chwilowo wprowadzający w błąd i nie powodujący konsekwencji. Niemoralne jest kłamstwo na temat przeszłości.	Branie forsy za popsute prog. nie jest uczciwe (tak robi M\$)! Sprzeczne jest z uczciwością rozdawanie gratisowego, wadliwego oprogramowania i wmawianie że to jakaś wzniosła idea. A tak naprawdę ten soft spowalnia pracę, ciągle denerwuje i psuje umysł na wiele sposobów.	NIE!
10. Solidność!!!	Moralne jest doprowadzanie projektów do końca jaki u odbiorcy nie generuje negatywnych uczuć. Zaleca się by wynik końcowy budził podziw i zachwyt. Nie moralne jest „odrabianie pańszczyzny” i wykonywanie zadań „byle jak”.	Tworzenie gratisowego, wadliwego oprogramowania jest zaprzeczeniem solidności!	NIE!

11. Braterstwo!!!	Moralna jest postawa koleżeńska. Nie moralne jest pomijanie lub szykanowanie innych ludzi. Moralne przestrzeganie zasady „Walczę o człowieka a nie z człowiekiem!”. Niemoralne jest pogrążanie ludzi przez wrabianie w łamanie jakiegoś tajnego lub niemoralnego prawa.	Tworzenie gratisowego, wadliwego oprogramowania przez programistów-sprzedawczyków jest zaprzeczeniem braterstwa!	NIE!
12. Droga i wdrożenia!!!	Moralne jest „zrobienie wszystkiego co możliwe” w celu pełnej realizacji proj. Wtedy mówimy, że droga do celu była moralna. Niemoralne są brak inicjatywy i zaniechania w wykonywaniu obowiązków, bo to jest sabotaż projektu.	Tworzenie gratisowego, wadliwego oprogramowania to droga do piekła! Nawet mimo wdrożeń.	NIE!
13. Sukcesy naukowe i kompetencje!!!	Moralne jest zdobywanie kol. tytułów naukowo-technicznych, oraz naukowe rozwiązywanie zagadek Naszego Wszechświata oraz inżynierskie rozwiązywanie problemów technicznych. Aby legalnie używać danej technologii trzeba na to uczciwie zasłużyć solidnie ucząc się obsługi i uczciwie pracując by legalnie tą technologię wytworzyć lub kupić.	Trudno powiedzieć coś o kompetencjach naukowych tych szatańskich twórców popsutych prog.	BRAK!

14. Kultura osobista!!!	Moralne jest panowanie nad swoim gniewem i nad swoimi żądzami. Moralne jest gonienie za legalnie uzyskiwaną wiedzą uzyskiwaną w drodze „białego wywiadu”, bo to rozwija intelekt. Niemoralne jest szafowanie wyrokami we wściekłości, bo to bezpośrednia droga do niesprawiedliwości i okrucieństwa, czyli zbrodni. Niemoralne jest gonienie za kradzionymi informacjami i za kradzionymi technologiami. Oraz: Niemoralne jest gonienie za despotyczną władzą, za brudnymi pieniędzmi, za seksem, za alko i narko bo taka gonitwa to kwintesencja pasożytnictwa.	Tworzenie gratisowego, wadliwego oprogramowania to brak kultury osobistej!	NIE!
15. Kultura techniczna!!!	Moralne jest wdrażanie pomysłów racjonalizatorskich i tworzenie udoskonalonych narzędzi koniecznych do dalszej pracy. Niemoralne jest odrzucanie pomysłów racjonalizatorskich.	Tworzenie gratisowego, wadliwego oprogramowania to brak kultury technicznej!	NIE!
16. Kultura ekonomiczna i własne utrzymanie!!!	Moralne jest przestrzeganie w interesach zasady „wygrany-wygrany”. Niemoralne jest łamanie praw ekonomii (tak jak to było dawniej w PRL i obecnie w UE). Moralna jest praca na własne utrzymanie.	Chęć zniszczenia umysłów innych ludzi to zaprzeczenie zasady „wygrany-wygrany”! Tworzenie gratisowego, wadliwego oprogramowania to brak kultury ekonomicznej!	NIE!

Ponieważ 13 jest na NIE! i 3 są na BRAK to znaczy, że tworzenie popsutego oprogramowania jest jednoznacznie niemoralne.

6.3 Analiza znanych faktów w świetle jednego, wybranego Wsk. Moralnej Popr.

Przeanalizuję moralną s. tworzenia popsutego oprogramowania w świetle Wsk. Moralnej Popr. zwanego Kultura Techniczna:

Ponieważ na bezrobociu mam sporo czasu opracowałem własną teorię tworzenia prog. nieuprzywilejowanych opartych o wtyczki [arch-prog-nieup]. Mając ten zbiór zasad przeanalizowałem kilka najczęściej używanych przeze mnie prog. dostarczanych razem z sys. op. Analizowałem kod prog. VLC wer. 3.0.9.2, KDE Krusader wer. 2.7.2 (Peace of Mind) i Qt Creator wer. 4.11.0, które są rozprowadzane razem sys. op. Ubuntu 20.04.

Moje analizy wskazują jasno, że te prog. są tworzone byle jak. W zasadzie to jakiś cud, że działają one stabilnie (a tak nie było przed 2010r.).

W badanych prog. są ograniczenia w ich rozszerzaniu: W przypadku KDE Krusader nie można tworzyć wtyczek. Natomiast w przypadku Qt Creator nie ma dok. jak te wtyczki tworzyć. Po za tym przyjęta w Qt Creator koncepcja, że „wszystko jest wtyczką” wydaje się przekomplikowana i ograniczająca programistów wtyczek.

Sam kod też budzi zastrzeżenia np. powszechny jest brak separacji od używanych bibl. - nie stosuje się przezywania typów, co dawniej było b. dobrym i częstym standardem.

Szczytem głupoty jest kod prog. VLC: mimo że kodeki VLC są zakodowane jako bibl. w przenośnym j. C, i mimo że istnieją bibl. wieloplatformowe takie jak darmowe Qt, albo SDL, to zamiast wyboru jednej z nich i konsekwentnego jej użycia, założono wiele odrębnych repozytoriów z kodem dla każdej platformy osobno. W chwili pisania (w d. 2024-09-13, pią.) są to repozytoria: Linuks/Widnows: github.com/videolan/vlc, MacOS: git.videolan.org/vlc.git, IOS: github.com/videolan/vlc-ios, Android: github.com/videolan/vlc-android. Nie ściągałem ich wszystkich na dysk, ale podejrzewam, że są to całkowicie odrębne prog. których wspólną cechą jest użycie bibl. współdzielonej w j. C z kodekami VLC. Oczywiście jest, że ktoś wyłożył grubą kapuchę by to wszystko wklepać i potem utrzymywać na chodzie. Takie szastanie kasą jest charakterystyczne dla proj. otwartego/wolnego oprogramowania. Dlatego całe te gratisowe oprogramowanie to jakiś „monkey-biznes”.

Mimo, że w KDE Krusader i w VLC dokumentacja była prowadzona w sys. Doxygen to w żadnym z nich nie można było z niej wygenerować dok. PDF z powodu błędów pojawiających się w trakcie ich generowania.

Od s. użytkowej programy te wymagały dodatkowych kliknięć, ale dzięki dostępnym kodom źródłowym można było to ograniczać modyfikując kod.

Te wszystkie wady wynikają z niskiej kultury technicznej twórców badanych kodów źródłowych wybranych prog. dostarczanych z sys. op. Ubuntu w wer. 20.04. Ta niska kultura wynika z kolei po prostu z braku dobrej woli u tych ludzi. Dlatego w świetle wsk. Kultura Techniczna świadome psucie oprogramowania jest niemoralne.

7 Podsumowanie

7.1 Rewelacje, zalety, wady i partactwa w udostępnianym oprogramowaniu

7.1.1 Rewelacje

1. To że te szatańskie sys. op. i prog. działają stabilnie.

7.1.2 Zalety

1. Dostępne kody źródłowe tych prog. (w sys. op. Linuks), które można modyfikować w celu ich naprawy.

7.1.3 Wady

1. BRAK

7.1.4 Partactwa

1. Sztuczne ograniczenia poziomu technicznego sys. op. dostępnych dla cywili (brak roz. porównywalnych do j. AREXX i brak możliwości pracy na prywatnych klastrach obliczeniowych znanych z Plan9);
2. Psucie umysłów przez utrudnienia w obsłudze prog. (wymuszanie zbędnych kliknięć);
3. Udostępnianie niekompletnego oprogramowania (opisane sed i grep);
4. Sztuczne utrudnienia w pracy programistów (opisane problemy ze blokadami prog. konsolowych uniemożliwiającymi użycie skryptach, problem z M\$ COM, powszechny brak dokumentacji, w tym brak dokumentacji jak tworzyć wtyczki do IDE takich jak Qt Creator, KDevelop i M\$ Visual Studio).

7.2 Wnioski

Obecnie udostępniane oprogramowanie z SZAP jest celowo przestarzałe i celowo popsute. Udostępnia się je nie po to by pomagać ludziom w życiu i w pracy, tylko po to by ich nękać i psuć ich umysły.

7.3 Czy problem jest organizacyjny czy jednostkowy?

Problem jest organizacyjny – to robią wielkie korpo, które najprawdopodobniej są wykonawcami instrukcji otrzymywanych od tajnej policji.

7.4 Czy problem jest trwały czy czasowy?

Problem jest trwały – co najmniej od 1994r.

7.5 Przewidywania na przyszłość

Dalej będą nam wciskać mierny soft z Ameryki.

7.6 Zalecenia na teraz

Używaj sys. op. Linuks – dziś on daje największe możliwości w pracy z komp. Prog. które często używasz przerabiaj tak by działały one normalniej. Jak jest to nie możliwe, to poszukaj czegoś co da się przerobić.

7.7 Zalecenia na przyszłość

Projektuj i programuj własne, nowoczesne i wygodne prog. Swoją codzienną pracę automatyzuj skryptami.

8 Licencja

Jest to licencja dotycząca tego dokumentu. Pliki wskazywane 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 czegokolwiek.
4. **Ograniczenia licencji nie dotyczą autora.**

9 Bibliografia

Bibliografia

commodore-s-w-wiki: , Commodore International, 2025, https://pl.wikipedia.org/wiki/Commodore_International

ibm-s-w-wiki: , IBM, 2025, <https://pl.wikipedia.org/wiki/IBM>

s-wiki: , Witaj w Wikipedii,wolnej encyklopedii, którą każdy może redagować., 2025,

https://pl.wikipedia.org/wiki/Wikipedia:Strona_g%C5%82%C3%B3wna

amigaos-s-w-wiki: , AmigaOS, 2025, <https://en.wikipedia.org/wiki/AmigaOS>

amigaos-influence-s-w-wiki: , AmigaOS, 2025,

https://en.wikipedia.org/wiki/Amiga#Influence_on_other_operating_systems

plan9-s-w-wiki: , Plan 9 from Bell Labs, 2025, https://en.wikipedia.org/wiki/Plan_9_from_Bell_Labs

Ideologia Geniuszy-Mocarzy: Jacek Marcin Jaworski, Ideologia Geniuszy-Mocarzy, 2025,

<https://energokod.pl/monografie/Ideologia%20Geniuszy-Mocarzy.pdf>

totalizm 8p, tom 1, s. 31: prof. Jan Pająk, Totalizm, 2004

arch-prog-nieup: Jacek Marcin Jaworski, Arch. Prog. Nieuprzywilejowanych, 2025, [http://energokod.pl/monografie/Arch.](http://energokod.pl/monografie/Arch.%20Prog.%20Nieuprzywilejowanych.pdf)

[%20Prog.%20Nieuprzywilejowanych.pdf](http://energokod.pl/monografie/Arch.%20Prog.%20Nieuprzywilejowanych.pdf)