

# Old Man GURU Magazine

*Wychodzi bardzo nieregularnie, kiedy wydaje mi się, że mam coś ciekawego lub pożytecznego do napisania...*

Numer 40/2013

22 sierpień 2013

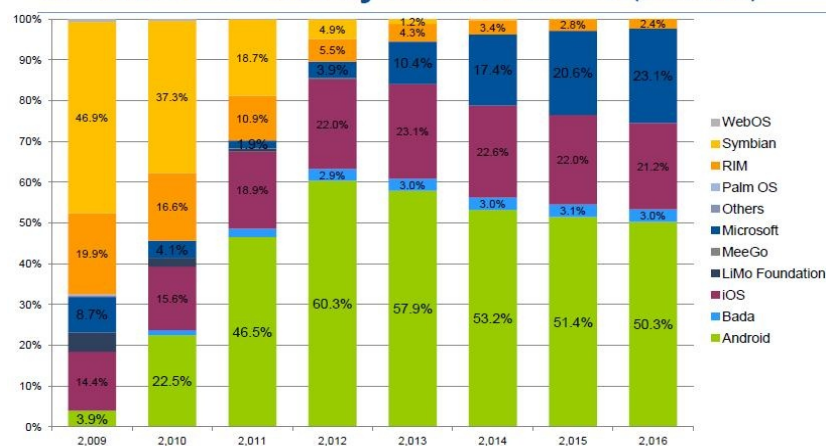
Pierwszy powakacyjny numer „Old Man Guru” został zainspirowany kilkoma rozmowami, które przeprowadziłem z osobami odpowiedzialnymi za pracę sporych systemów informatycznych. Byli to oczywiście obecni lub potencjalni klienci firmy ABA i jej Partnerów. We wszystkich tych rozmowach oczywiście przewijał się temat wykorzystywania Linuksa w instalacjach profesjonalnych, który wbrew lamentom niektórych organizacji jest coraz powszechniej stosowany.

Uwagi moich rozmówców były bardzo konkretne i dały mi sporo do myślenia:

Przed wszystkim moi szanowni rozmówcy podkreślali, że według ich rozeznania koszt wdrażania Linuksa to przede wszystkim koszt związany ze zdobywaniem określonych kompetencji w prowadzonych przez nich zespołach. Utrudnia to przede wszystkim wprowadzenie systemów otwartych jako środowiska pracy dla wielu przeciętnych użytkowników. Jeden z moich rozmówców powołał się na sukces Androida:

*„System Android zdobył przebojem rynek telefonów komórkowych. Potwierdzają to badania firmy Gartner. W ciągu 3 lat z ciekawostki (udział w rynku na poziomie 4%) stał się dominującym graczem z ponad 60% udziałem w 2012 roku. Jeśli nawet te dane (i przewidywania) są obarczone błędem to nie sposób zanegować faktu, że Android oparty o technologię Linuksa został powszechnie zaakceptowany przez użytkowników urządzeń przenośnych.”*

## Gartner Forecast Estimates Mobile OS Sales by Market Share (2009-2016)



Source: Gartner  
Forecast: Mobile Devices by Open Operating System, Worldwide, 2009-2016, 2Q12 Update

Gartner

*„Co spowodowało, że użytkownicy zaakceptowali Androida (czyli de-facto Linuksa)? - kontynuował mój rozmówca. **Przed wszystkim łatwość użytkowania i dostępność oraz prostota instalacji aplikacji! Niestety, te cechy nie charakteryzują standardowych dystrybucji Linuksa**”.*

Trudno mojemu rozmówcy odmówić racji. Przeciętnego użytkownika „smartphone” czy tabletu wcale nie obchodzi, jaki system operacyjny wykorzystuje jego urządzenie. Ważne, aby pracowało stabilnie i aby realizowało niezbędne użytkownikowi funkcje.

Drugim podnoszonym przez moich rozmówców problemem była powtarzalność rozwiązań:

*„Linux jest na pewno dojrzałym i sprawnym systemem operacyjnym. Sam używam Linuksa na swoim domowym komputerze. Jednak administrowanie siecią, w której mamy 600 stanowisk pracy to co innego. Tu nie można sobie pozwolić na zbyt częste zmiany oraz stosowanie różnych rozwiązań. Taki system musi wykorzystywać rozwiązania standardowe. Argument w rodzaju „w Linuksie możesz zrobić wszystko” w takim środowisku po prostu nie działa!”* stwierdził drugi z moich rozmówców.

Też trudno nie przyznać mu racji – moje doświadczenia z opracowania i wdrożenia oprogramowania opartego o system Linux dla ponad 15000 stanowisk pracy (2003 rok dla Ministerstwa Finansów) w pełni potwierdzają tę tezę. Również w mniejszych lokalnych instalacjach (np. kilkaset stanowisk w szpitalu) większe znaczenie ma stabilność i powtarzalność rozwiązania niż liczba możliwych do wykorzystywania funkcji. To, co jest zaletą dla entuzjasty komputerów (linuksowego „geeka”) staje się szybko przysłowiową „kulą u nogi” w codziennej praktyce działu IT profesjonalnej organizacji. Trudno przecież wymagać od użytkowników, aby zarządzali swoją stacją roboczą (niezależnie od tego, czy pracuje pod systemem Windows czy Linux) – dziś komputery służą po prostu do realizacji codziennych zadań powinny i być tak proste w użytkowaniu jak długopis i kartka papieru połączone z dobrze zorganizowaną kartoteką. Co prawda inny z moich rozmówców stwierdził, że posiadanie komputera służbowego najnowszej generacji jest w niektórych organizacjach wyznacznikiem pozycji w hierarchii służbowej (podobnie, jak model samochodu służbowego) – ale powoli takie postawy stają się coraz rzadsze (także na ulicy nowy model Mercedesa lub Porsche nie jest już taką atrakcją, jak kiedyś).

Z moich rozmów wynika kilka ważnych wniosków:

1. Najważniejsze cechy stanowiska pracy to stabilność i niezawodność. Te cechy decydują o wydajności pracownika oraz zapewniają mu komfort pracy,
2. Dla działów IT najistotniejsza jest tak zwana „serwisowalność”. Jeden z moich rozmówców stwierdził, że najlepszym rozwiązaniem redukującym do minimum czas reakcji serwisowej jest możliwość natychmiastowej wymiany stanowiska pracy na nowe, sprawne. Wymaga to oczywiście odpowiedniego oprogramowania systemowego oraz centralizacji składowania danych,
3. Kilku rozmówców podkreślało zalety wynajmu stacji roboczych – rozkłada to w czasie koszty, likwiduje konieczność organizacji logistyki serwisu oraz likwidacji i recyklingu wycofywanego sprzętu,

Wprowadzana obecnie konsolidacja serwerów oraz rozwiązania wirtualizacji (łącznie z wirtualizacją desktopu) umożliwiają spełnienie powyższych wymagań. Konieczne jest jednak odpowiednie przygotowanie oprogramowania stacji roboczej.

Przygotowując kolejną modernizację oprogramowania końcówki ABA-X4 (od 20 sierpnia jest to wersja 1.2.0) starałem się uwzględnić zgłaszane przez dyrektorów działów IT uwagi. Umożliwia to konsekwentna budowa modułarna systemu końcówki:

#### Podstawowa część systemu operacyjnego – Core System

Ta część systemu jest ładowana do pamięci RAM. Wykorzystano w tym celu standardowy mechanizm Linuksa – initrd (Initial RAM disk). Plik ten jest używany w standardowych dystrybucjach Linuksa w celu załadowania ostatecznej wersji systemu z dysku twardego.

W rozwiązaniu ABA-X4 zawiera on odpowiednio przygotowany i skompresowany ostateczny obraz głównego systemu plikowego (root filesystem), który jest ładowany do pamięci RAM. W odróżnieniu od standardowych dystrybucji Linuksa system ten jest używany jako podstawowy przez cały czas pracy końcówki. Plik ten (core.gz) jest stosunkowo niewielki (ok. 6 MB) i wraz z jądrem systemu Linux (ok. 2,5 MB) zajmuje nawet po rozkompresowaniu niewiele miejsca w pamięci RAM:

Filesystem	Size	Used	Available	Use%	Mounted on
rootfs	1.8G	30.1M	1.7G	2%	/

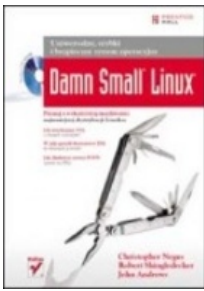
Takie rozwiązanie (stosowane również w rozwiązaniach typu LiveCD) zapewnia:

- Bezpieczeństwo pracy – nawet w przypadku uszkodzenia - przypadkowego lub celowego (np. przez złośliwy program) systemu plikowego root po twardym restarcie końcówki nastąpi ponowne władowanie poprawnej wersji systemu,
- Szybkość i wydajność – dostęp do danych w pamięci RAM jest znacznie szybszy niż w przypadku twardego dysku lub innych pamięci masowych (SSD, pamięci FLASH itp.),
- Uniwersalność – Core System może być załadowany do RAM zarówno z dowolnego nośnika (PenDrive, Karty SD, CF lub podobne, HDD, SSD...) jak i z sieci (PXE boot),
- Łatwość uaktualnień – zarówno jądro, jak i plik core.gz można łatwo wymienić na nowe wersje. Niewielka objętość tych plików umożliwia szybką realizację „update” także przez sieć – nawet rozległą.

Projektując końcówkę ABA-X4 założyłem, że jądro systemu oraz plik core.gz (initrd) powinny być w miarę możliwości jednakowe dla wszystkich końcówek. Zdecydowanie ułatwia to serwis oraz zarządzanie kopiami awaryjnymi (wystarczy jedna nawet na kilka tysięcy końcówek).

Jak już wspomniałem powyżej dla użytkownika najważniejsze są programy, których używa, realizacja dostępu do usług i serwerów sieciowych – a nie sam system operacyjny. W klasycznych systemach (również dystrybucjach Linuksa) oprogramowanie użytkowe jest po prostu instalowane na dysku komputera. Programy instalacyjne tworzą odpowiednie struktury katalogowe oraz umieszczają w nich wymagane pliki, które stają się w ten sposób częścią systemów plikowych, z których korzysta system operacyjny.

Niektórzy dostawcy (słynna sprawa Explorera) integrowali również programy użytkowe z systemem operacyjnym w taki sposób, że stawały się one jego nieodłączną częścią. Na szczęście te praktyki już chyba należą do przeszłości.



Oprogramowanie ABA-X4 opracowano wykorzystując koncepcję Roberta Shingledeckera, programisty o ogromnym doświadczeniu (napisał między innymi dla swego domowego komputera kompilator języka COBOL) w budowie małych dystrybucji (Minix, Coherent Unix, Damn Small Linux oraz Tiny Core Linux), współautora książki „Damn Small Linux” (DSL) wydanej także w Polsce przez „Helion”.

Ze względu na nieporozumienia (widać, że nie to tylko polska specjalność) w zespole DSL Robert „odszedł na swoje” i rozpoczął projekt TinyCore Linux.

Wróćmy jednak do techniki:

Każdy (porządnie napisany!) program komputerowy zawiera dane statyczne (binaria, biblioteki itp.) oraz dynamiczne (dane przejściowe, pliki konfiguracyjne itp.). Ideą wykorzystaną przez TinyCore jest separacja danych statycznych i dynamicznych. Jeśli np. w naszym systemie instalujemy dowolny program, to wprowadzamy głównie dane statyczne, które dla wszystkich realizowanych instalacji będą takie same. Dopiero konfiguracja oraz użytkowanie systemu powodują powstawanie danych dynamicznych, które są różne dla różnych komputerów.

Efekt ten wykorzystano w dystrybucji TinyCore opracowując tak zwane rozszerzenia (extensions). Pomimo, że plik rozszerzenia (np. z LibreOffice, Firefoksem itp.) zawiera informacje o rozkładzie katalogów, wymaganych zależnościach itp. nie należy tego rozwiązania traktować podobnie jak plików .rpm lub .deb! Rozszerzenia nie są bowiem instalowane, tylko zapisywane na nośniku (w katalogu tce/optional):

```
...
-rw-rw-r--  1 tc      staff    12955648 May 10 17:39 perl5.tcz
-rw-rw-r--  1 tc      staff         44 May 10 17:39 perl5.tcz.md5.txt
...
-rw-rw-r--  1 tc      staff    274432 Aug 21 15:36 zenity.tcz
-rw-rw-r--  1 tc      staff         23 Aug 21 15:33 zenity.tcz.dep
-rw-rw-r--  1 tc      staff         45 Aug 21 15:36 zenity.tcz.md5.txt
```

a następnie montowane w trybie „tylko do odczytu”:

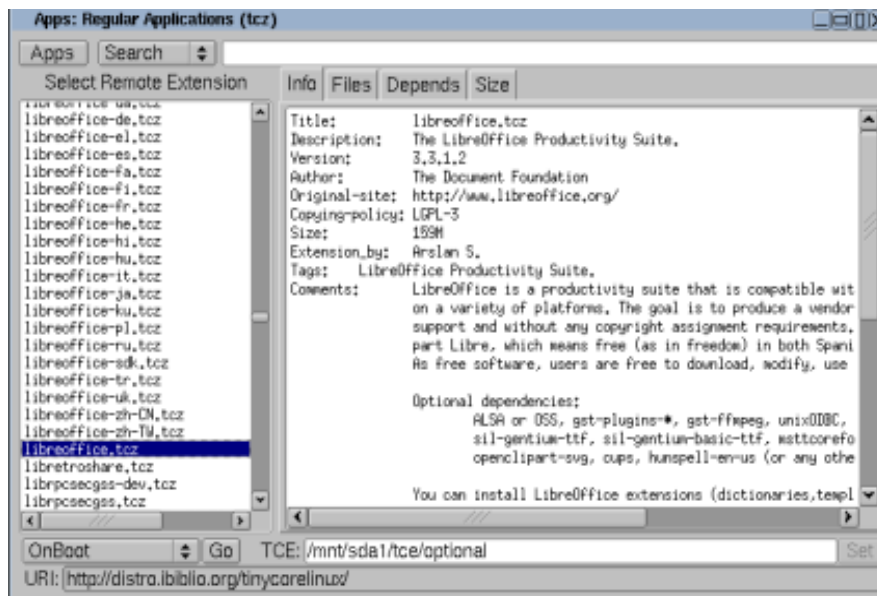
```
mount |grep perl5
/dev/loop6 on /tmp/tcloop/perl5 type squashfs (ro,relatime)
```

i udostępniane w postaci linków symbolicznych:

```
lrwxrwxrwx  1 root    root          42 Aug 22 08:33 perl5.14.1 -> /tmp/tcloop/perl5/usr/local/bin/perl5.14.1
```

Takie podejście wymaga stworzenia wielu urządzeń typu „loop” - nawet w prostym terminalu ABA-X4 może ich być kilkadziesiąt, a nawet w bardziej złożonych wersjach znacznie powyżej 100. Nie jest więc celowe tworzenie w ten sposób bardzo złożonych systemów, jednak w przypadku tworzenia oprogramowania dla terminala sieciowego, cienkiego klienta lub nawet stacji roboczej, wyświetlacza „digital signage”, różnych „set-top boxów” zalety znacznie górują nad ograniczeniami.

- Uszkodzenie programu udostępnionego w postaci rozszerzenia nie jest w praktyce możliwe. Oczywiście, uszkodzenie takie nastąpi w wyniku awarii nośnika, jednak systemy plikowe typu squash wersji stosowanej w TinyCore (a więc i w ABA-X4) umożliwiają jedynie odczyt bez możliwości zapisu. Stanowi to znakomite zabezpieczenie przed uszkodzeniem oprogramowania.
- Rozszerzenia mogą być udostępniane w różny sposób – również np. za pomocą NFS.
- Zarządzanie oprogramowaniem jest bardzo proste – i sprowadza się do pobrania pliku rozszerzenia (najczęściej z repozytorium sieciowego).
- Ponieważ rozszerzenia zawierają spis niezbędnych zależności (np. listę niezbędnych bibliotek) są one pobierane i udostępniane automatycznie przez program ładujący rozszerzenia.
- Obsługa rozszerzeń (sprawdzenie listy, zależności, montowanie i udostępnianie) odbywa się całkowicie automatycznie podczas każdego startu urządzenia.
- Rozszerzenia są udostępniane w sieci (lista obejmuje kilkaset modułów). Rozszerzenia można również tworzyć we własnym zakresie.



Pozostają pliki konfiguracyjne, które oczywiście mogą być różne dla każdej stacji roboczej (terminala). Są to zarówno pliki systemowe (zestawy zmiennych konfiguracyjnych – np. plik zawierający konfigurację sieci ifcfg-eth0), pliki konfiguracji środowiska użytkownika (.profile, .xsession itp.) jak i pliki generowane dla użytkownika np. przez przeglądarkę (katalog .mozilla w katalogu domowym użytkownika).

W oprogramowaniu ABA-X4 takie pliki są umieszczane w 2 katalogach - /home oraz /opt.

Katalog /opt przeznaczony jest dla systemowych plików konfiguracyjnych oraz skryptów uruchomieniowych, za katalog /home dla plików użytkownika systemowego terminala.

Zabezpieczenie tych danych jest realizowane w systemie ABA-X4 na dwa sposoby:

- Dla prostych końcówek, które są wykorzystywane jedynie do połączeń z serwerami usług terminalowych MS Windows, Linux/Unix itp. oba powyżej wymienione katalogi są zapisywane we wspólnym archiwum.  
Ponieważ przy czystej pracy terminalowej żadne dane nie są zapisywane na terminalu zapis archiwum jest konieczny jedynie w przypadku dokonywania zmian w konfiguracji terminala (rozdzielczość ekranu, ustawień sieciowych itp.).  
Archiwum jest rozpakowywane podczas każdego startu terminala – dopiero wówczas powstają katalogi /home i /opt oraz umieszczane są w nich odpowiednie pliki.  
Sposób ten znakomicie zdaje egzamin w przypadku, gdy archiwum jest stosunkowo niewielkie, ponieważ konieczność jego rozpakowania wydłuża czas startu systemu.  
Zaletą jest oczywiście prostota i łatwe wdrożenie systemu zdalnej konfiguracji nawet dla wielu setek końcówek. Ponieważ w archiwum konfiguracyjnym zawarta jest cała konfiguracja końcówki – wystarczy więc załadowanie go za pośrednictwem sieci i wszystkie ustawienia końcówki zostaną zmienione. Ułatwia to też wymianę uszkodzonej końcówki redukując do minimum czas jej skonfigurowania odpowiednio do realizowanej funkcji.
- W stacjach roboczych, na których są realizowane bardziej złożone zadania, a w szczególności następuje lokalne przetwarzanie danych ABA-X4 wykorzystuje inny sposób. Katalogi /home i /opt są trwale zapisywane na lokalnym nośniku lub udostępniane w sieci np. za pomocą NFS lub SMB (CIFS). Czas uruchomienia systemu przy takim rozwiązaniu nie wzrasta wraz ze wzrostem objętości plików użytkownika (szczególnie kłopotliwe są przeglądarki sieciowe – katalog .mozilla w katalogu domowym może łatwo zająć kilkadziesiąt MB). Nie wyklucza to oczywiście tworzenia archiwów tych katalogów, lecz realizowane są one osobno. W takim przypadku nie jest stosowany automatyczny system ich odtwarzania, lecz może to być zrealizowane w każdej chwili na polecenie administratora sieci.

O ile jednak katalog /opt zawierający konfigurację systemową może być udostępniany w trybie tylko do odczytu (np. jak osobny system plikowy), to jeśli chcemy skorzystać z zaawansowanych funkcji przeglądarki (np. Firefox) to katalog /home musi być dostępny do zapisu.

Funkcje, jakich wymaga się od końcówki determinują sposób postępowania z plikami zmiennymi. Należy dokonać optymalnego wyboru biorąc pod uwagę czas uruchamiania końcówki oraz potrzebę lokalnego trwałego przechowywania plików (np. historii przeglądarki). W każdym jednak przypadku należy zachować najważniejsze zalety systemu – brak potrzeby jakiegokolwiek obsługi lub konfiguracji stanowiska pracy przez użytkownika. Działania użytkownika powinny się ograniczać jedynie do włączania i wyłączania stacji roboczej (końcówki), która powinna realizować połączenia z serwerem (serwerami) całkowicie automatycznie, zaś natychmiastowe wyłączenie lub zanik zasilania nie powinny skutkować uszkodzeniem oprogramowania.

Oczywistą przewagę ma tu pierwsze rozwiązanie wykorzystujące odtwarzanie katalogów /home i /opt podczas uruchamiania terminala. Ponieważ użytkownik nie dokonuje zmian w archiwum (jest to zastrzeżone jedynie dla administratora) może być ono odpowiednio zabezpieczone i nie jest wymagane jego tworzenie przy kończeniu pracy na terminalu.