

Autor:	mgr inż. Przemysław Adam Śmiejek
Licencja:	The GNU Free Documentation License <a href="http://www.gnu.org/licenses/licenses.html#FDL">http://www.gnu.org/licenses/licenses.html#FDL</a> <a href="http://gnu.org.pl/text/GFDL-pl.html">http://gnu.org.pl/text/GFDL-pl.html</a>
Data:	Listopad 2003
Wersja:	0.1

## 1. Jak powstał GNU/Linux

---

W połowie lat 80, zniechęcony komercjalizacją świata informatycznego, Richard Matthew Stallman (znany jako RMS) postanowił założyć fundację Free Software Foundation i opracować ideę Wolnego Oprogramowania (Free Software). Jego fundacja zajęła się tworzeniem całkowicie ogólnodostępnego systemu GNU<sup>1</sup> będącego funkcjonalnym odpowiednikiem komercyjnego Uniksa. Prace rozpoczęto od stworzenia zestawu kompilatorów<sup>2</sup> gcc (GNU Compiler Collection) oraz zestawu różnych podstawowych narzędzi systemowych. Na sam koniec zostawiono sobie budowę jądra systemu, które nazwano Hurd. Jądro jest centrum działania każdego systemu operacyjnego. Odpowiedzialne jest ono za sterowanie działaniem sprzętu, komunikacją ze sterownikami i programami.

### 1.1. Powstaje Linux

Nadszedł rok 1991. Młody student z Finlandii, Linus Torvalds, wykorzystując kompilator gcc oraz założenia systemu GNU zbudował jądro, które nazwał Linux. Jądro to powstało jako hobby młodego Linusa i zostało upowszechnione na grupach dyskusyjnych w formie ciekawostki. Szybko jednak społeczność akademicka podchwyciła pomysł i poczęto go rozwijać przy pomocy ochotników z całego świata. Dzięki uzupełnieniu o narzędzia ze stajni GNU zaczęło stanowić spójny system, nazywany GNU/Linux. Fundacja RMS-a zajęła się rozwojem narzędzi systemowych, a Linus pracował nad swoim jądrem współpracującym z tymi narzędziami. Z niewiadomych przyczyn, popularność w świecie zdobyła nazwa Linux i tym mianem często określa się cały system operacyjny, oparty o stworzone przez fińskiego informatyka jądro zdominując o rodowodzie reszty narzędzi.

### 1.2. Paczkowanie w „dystrybucje”

Dla komputerowego guru, zdobycie zestawu oprogramowania, połączenie go z jądrem, zainstalowanie i uruchomienie nie stanowi problemu. Jednakże sporo osób

---

<sup>1</sup> GNU to rekurencyjny skrót od GNU is Not Unix

<sup>2</sup> Kompilator to narzędzie tłumaczące kod źródłowy programu napisany przez programistę na język zrozumiały przez komputer. W uproszczeniu można powiedzieć, że kompilator umożliwia tworzenie oprogramowania.

pragnie systemu instalowanego i uruchamianego w wygodny sposób, połączonego i przetestowanego pod kątem spójnego działania. Tak narodziły się „dystrybucje”, czyli zebrane w jedną całość elementy systemu GNU/Linux. Pierwszą „dystrybucją” był Slackware. Potem pojawiły się Debian i RedHat. Za nimi podążały inne. Ponieważ każdy może zebrać zestaw Wolnego Oprogramowania i złożyć swój własny system, „dystrybucji” na świecie jest obecnie bardzo wiele<sup>3</sup>.

### 1.3. Nie samym GNU żyje świat

Popularność GNU/Linuksa sprawiła, iż poczęto tworzyć różnorakie oprogramowanie działające pod tym systemem. Część osób dołączała do projektu GNU, inni stosowali co prawda licencję GNU GPL, jednakże z samym Free Software Foundation się nie wiąźali. Jeszcze inni twórcy postanowili nawet wykorzystywać inne sposoby licencjonowania swoich produktów. Część tego oprogramowania zaczęła trafiać do „dystrybucji”, które stały się już nie tylko zbiorem oprogramowania GNU/Linux, ale samodzielnymi systemami operacyjnymi, zawierającymi różne składniki, także pochodzące spoza projektu GNU. Stąd też niektórzy negują ten przedrostek w nazwie twierdząc, że przecież Mandrake czy Aurox to nie GNU/Linux, ale coś więcej. Z uwagi jednak na olbrzymie podobieństwo do siebie współczesnych systemów opartych o GNU/Linuksa oraz brak sensownej nazwy obejmującej całość, wciąż używa się nazw takich jak „dystrybucja Linuksa”, „Linux” czy też „GNU/Linux” na określenie całej rodziny systemów. Tej nazwy będę więc używał w dalszej części.

## 2. Czym charakteryzuje się GNU/Linux

---

GNU/Linux to system o budowie charakterystycznej dla systemów uniksowych.

### 2.1. Podstawowe elementy

Kompletny system (Debian, Mandrake, Auroks itp.) składa się z jądra o nazwie Linux, bibliotek systemowych (główną rolę pełni glibc), powłoki tekstowej (najczęściej jest to bash) służącej do komunikacji z systemem przy pomocy poleceń wpisywanych z klawiatury, systemu graficznego X Window System, oprogramowania zarządzającego obsługą aplikacji graficznych (KDE, Gnome, WindowMaker itp.) oraz niezliczonej liczby ważnego i mniej ważnego oprogramowania, takiego jak serwery (WWW, poczta, DNS itp.) lub klienty (programy pocztowe, przeglądarki). Znajdziemy w nim również sporo dodatkowego oprogramowania, w tym także edukacyjnego.

To, jakie oprogramowanie przychodzi razem z systemem, zależy od autorów danej wersji. Przykładowo Aurox mieści się na 3 płytach, Mandrake na 4, Debian na 6 lub 7. W razie czego, zawsze można doinstalować interesujące pakiety, nawet jeśli nie znalazły się na płytach dystrybucyjnych systemu.

<sup>3</sup> Zainteresowanych odsyłam do serwisu [www.distrowatch.com](http://www.distrowatch.com) zajmującego się indeksowaniem najpopularniejszych „dystrybucji”

## 2.2. Administracja

Administracja systemem następuje głównie przy użyciu trybu tekstowego. Nie jest to może ładne, kolorowe i na czasie, ale bardzo skuteczne. Mamy do czynienia z prostymi plikami tekstowymi, które można łatwo zarchiwizować czy też przenieść z innej, już dobrze działającej maszyny.

Dla osób lubiących graficzne konfiguratorry tworzone są narzędzia umożliwiające graficzne zarządzanie systemem. Elementy te są szczególnie dobrze rozwinięte w takich systemach jak Mandrake czy Aurox, które skierowane są do przeciętnego użytkownika. Należy jednak pamiętać, że nie zawsze działają one w 100% sprawnie i zazwyczaj obejmują jedynie część możliwości konfiguracyjnych. Programy ze środowiska Wolnego Oprogramowania są bowiem silnie konfigurowalne i mocno personalizowalne. Każda opcja i każdy element działania dające się modyfikować, umieszczane są w plikach sterujących działaniem programu.

Ogromną pomocą dla administratora są również tzw. logi systemowe, czyli pliki z zapisem przebiegu pracy konkretnych programów. Chcąc wiedzieć dlaczego poczta nie dochodzi do odbiorcy, dlaczego strona WWW się nie wyświetla, zaglądamy do zapisków prowadzonych przez konkretne serwery. Zaglądamy tam niekoniecznie z powodu awarii. Nawet jeśli wszystko działa idealnie, logi dostarczają na przykład cennych informacji na temat ilości odwołań do danej strony WWW, czy miejsc, z których odwiedzali serwer goście, dzięki czemu możliwe jest łatwe i wygodne tworzenie wszelakich statystyk czy zestawień.

## 2.3. Środowiska graficzne

Jak już wspomniałem, za działanie trybu graficznego odpowiedzialny jest X Window System. Sam jednak na wiele się nie nada typowemu użytkownikowi, gdyż zajmuje się on głównie obsługą działania grafiki. Dlatego istnieją różne środowiska graficzne, takie jak KDE, Gnome, WindowMaker, IceWM udostępniające obsługę plików, konfigurację parametrów wyświetlania obrazu, biblioteki graficzne dla programistów itp. Środowiska te sterują wyglądem okienek, uruchamianiem programów. W dużym uproszczeniu można rzec, że są odpowiednikiem graficznego interfejsu użytkownika z Windows<sup>4</sup>. Istotne jest jednak to, że jest ich wiele i mają różne cechy. Są rozwiązania takie jak KDE, które upodobniają się do Windows, ale także całkowicie odmienne, jak WindowMaker czy FluxBox. Niektóre z nich wymagają sporej ilości zasobów komputera (pamięci, mocy obliczeniowej), a niektóre działają nawet na stareńkim 386.

## 2.4. Dostępny kod źródłowy

Praktycznie wszystkie elementy dostępne są z kodem źródłowym, gdyż jest to cecha wręcz wymagana przez licencję GNU GPL. Kod źródłowy jest to zapis zestawu instrukcji i funkcji, sformułowany w określonym języku programowania (C, C++, Perl,

---

<sup>4</sup> W uproszczeniu, gdyż w przypadku Windows mamy do czynienia z budową, w której wszystko połączone jest w jedną całość i brak jest różnorodności. Co innego w środowisku X-Window, gdzie mamy do czynienia z modułową budową i różne elementy można wymieniać i zastępować innymi.

Python itp.). Programista tworząc program, pisze właśnie kod źródłowy, który dopiero potem przerabia (kompiluje) na postać zrozumiałą dla komputera.

Dostęp do kodu źródłowego niesie za sobą wiele korzyści. Łatwo jest poznawać budowę programu, łatwo nanosić własne poprawki. Łatwiej też stwierdzić, czy przypadkiem program za naszymi plecami nie szpieguje i nie wysyła jakiś tajnych informacji do producenta czy też innych osób.

## 2.5. Szyty na miarę

Jak zapewne już wiele osób się zorientowało, ważną cechą systemów z rodziny GNU/Linux jest modułowa budowa. Każdy może dobrać te składniki świata Wolnego Oprogramowania, które mu odpowiadają i poskładać z nich, niczym z klocków, system służący określonym celom. Do tego, dzięki dostępności kodu źródłowego, można w razie potrzeby modyfikować już istniejące programy tak, aby spełniały nawet bardzo specyficzne wymagania.

## 2.6. Skalowalny i modyfikowalny

Dzięki wolności, jaką niesie Wolne Oprogramowanie, nie jesteśmy skrzepowani ograniczeniami narzuconymi przez licencję. Przykładowo, sieć zakładowa składa się z 10 stanowisk. Pewnego dnia zatrudnia się 11 pracownika. „Proszę bardzo, panie Nowak, tu jest pana kabelek sieciowy. Opps... Mamy licencję tylko na 10 podłączeń do serwera.” Taka sytuacja nie występuje w przypadku Wolnego Oprogramowania. Rozbudowujemy infrastrukturę, instalujemy nowe oprogramowanie, wszystko bez stresów, bez wątpliwości, czy posiadamy wystarczającą ilość licencji, czy licencja zezwala na takie lub inne wykorzystanie, bez konieczności rejestracji u producenta przy każdej zmianie konfiguracji komputera.

## 2.7. Może działać bez instalacji

Coraz modniejsze stają się systemy działające bezpośrednio z płyty CD. Zawierają one system, który sam się uruchamia, wykrywa dostępne elementy komputera i działa bez ingerencji w dysk twardy. Takie wersje używane są zarówno jako narzędzia ratunkowe wykorzystywane w czasie awarii, ale także jako system do działań w miejscu, gdzie nie można lub nie ma czasu na typową instalację. Przykładowo pakiet Linux EduCD to działająca z płyty wersja zawierająca między innymi pakiet biurowy, przeglądarkę internetową oraz zestaw oprogramowania edukacyjnego. Tak przygotowana płytka może służyć podczas lekcji, nawet jeśli opiekun pracowni jest człowiekiem zajęczego serca i boi się instalacji pełnoprawnego GNU/Linuksa,

## 3. Dlaczego Wolne Oprogramowanie w szkole?

---

Nierzadko można spotkać się z głosami sprzeciwu mówiącymi „Po co Linux? Przecież wszędzie jest Windows, uczeń musi mieć styczność z Windows”. Tak to prawda. Jednakże czy fakt, że wszędzie jest Windows to zdrowy fakt? Postaram się wyjaśnić dlaczego należy odpowiedzieć NIE na to pytanie<sup>5</sup>.

### 3.1. Inwazja „maluchów”?

Co byśmy powiedzieli, gdyby na naszych ulicach królowały wyłącznie produkty FSM-u zwane popularnie „maluchami”, albo dowolne inne, lecz wyłącznie jednej marki pojazdy? Sama myśl o takiej ewentualności budzi w nas naturalny sprzeciw. Czemu więc godzimy się na fakt jedynie słusznego systemu operacyjnego oraz oprogramowania biurowego? Dlaczego prawie wszyscy używamy programów takich jak Internet Explorer czy Outlook Express, choć powszechnie wiadomo, że są dziurawe, niebezpieczne, grożą nam wirusami. Używamy produktów sprzed kilku lat, nie łatanych, niepoprawianych, pisanych w chaosie i naszpikowanych błędami. Dlaczego? Dlatego, że Microsoft idealnie przemyślał marketing. Podane na tacy produkty, nawet jeśli liche i koślawe, będą używane przez osoby niezorientowane.

Systemy i rozwiązania alternatywne dla produktów Microsoftu są czasami lepsze, czasami gorsze. Są po prostu inne. I nasz uczeń powinien wiedzieć, że są, jak wyglądają. Powinien umieć się nimi posłużyć. Przecież nie jesteśmy płatnym trenerem Microsoftu. Dlaczego więc mamy uczyć na lekcjach obsługi Worda? Czy nie powinniśmy raczej uczyć posługiwania się dowolnym edytorem tekstu? Dobrze nauczony młody człowiek poradzi sobie zarówno z Wordem, jak i z Writerem z pakietu OpenOffice lub KWordem z pakietu KOffice.

### 3.2. Oszczędni się bogacą

Wolne Oprogramowanie jest najczęściej oprogramowaniem dostępnym darmowo lub za niewielką opłatą, bez ograniczeń w ilości instalacji. Ważnym powodem są więc oszczędności jakie daje zastosowanie darmowego oprogramowania — nie we wszystkich miejscowościach szkoły stać na zakup licencji dla każdego stanowiska. Wolne Oprogramowanie daje możliwość szkołom lub uczniom kopiowania i redystrybucji używanego oprogramowania, tak więc na przykład szkolny system może być skopiowany na wszystkie komputery we wszystkich szkołach oraz zabrany przez uczniów do domu. Zamiast wymagać<sup>6</sup> od ucznia posiadania kosztującego około 2000zł pakietu Microsoft Office, można zasugerować wykorzystanie całkowicie darmowego OpenOffice.

Owszem, świadomy zagrożenia Microsoft oraz inne korporacje próbują oferować tanie wersje edukacyjne. Jednakże to celowe działanie nie mające na celu bynajm-

---

<sup>5</sup> Znaczna część poniższego tekstu i argumentów bazuje na wypowiedzi *Richarda M. Stallmana* oraz na artykule z serwisu *Linux w Polsce* (<http://www.linux.pl/?id=news&show=575>).

<sup>6</sup> W polskich realiach: Namawiać do kradzieży

niej ulżenia młodzieży. Wszak mówi ludowe przysłowie: „Czym skorupka za młodu nasiąknie...”.

### 3.3. Nieutrudniona wymiana technologii

Wolne Oprogramowanie dostępne jest wraz z kodem źródłowym. Dzięki temu wiadomo jak zostało zbudowane, jak działa. Kiedy uczniowie osiągną wiek dojrzały część z nich chce się dowiedzieć jak najwięcej o swoim komputerze i jego oprogramowaniu. Dostęp do kodu źródłowego stanowi pomoc w nauce. Komercyjne oprogramowanie nie pozwala na poznanie swego kodu, zdaje się mówić: „*To co chcesz poznać jest sekretem — nauka jest zabroniona!*”. Dlatego środowiska akademickie, naukowe tak chętnie sięgają po Wolne Oprogramowanie i tak wielki wkład w jego rozwój wnoszą. Wiadomo bowiem, że opracowywane technologie będą służyły kolejnym grupom zainteresowanych i będą dalej rozwijane (o ile okażą się interesujące).

### 3.4. Rodzimy fachowiec

Administrując systemem takim jak Windows, czujemy się jak gorsza odmiana człowieka. Otrzymujemy gotowy produkt, który nie wiadomo jak działa, dlaczego coś się psuje. Musimy zdać się na łaskę i niełaskę korporacji zza oceanu. Natomiast system oparty o Wolne Oprogramowanie, z dostępnym kodem źródłowym, może być poprawiany, a nawet rozwijany przez fachowców dostępnych na miejscu. Inwestycja w Wolne Oprogramowanie to inwestycja w stanowiska pracy dla lokalnych programistów, administratorów, wdrożeniowców, serwisantów. Dlatego właśnie administracja publiczna w krajach Europy Zachodniej sięga po systemy z otwartym kodem. Dlatego władze Chin zleciły opracowanie systemu informatycznego opartego o Wolne Oprogramowanie. Dzięki temu pracę i zyski mają nie pojedyncze jednostki w USA, ale obywatele konkretnego kraju.

Wdrażając uczniów do pracy z produktami ze świata Wolnego Oprogramowania, przygotowujemy ich więc technicznie i psychologicznie do pracy nie na rzecz obcych państw, ale dla własnego dobra.

## 4. Licencjonowanie oprogramowania

---

Wokoło Wolnego Oprogramowania i licencji z nim związanych, narosło wiele nie-domówień i niejasności. Spróbuję więc scharakteryzować główne aspekty tego rodzaju licencjonowania.

W powszechnym użyciu mamy najczęściej następujące licencje:

- *Komercyjna*. Określa zasady wykorzystania produktu, narzuca swoje wymagania i zezwala na używanie programu dopiero po uiszczeniu opłat. Prawa autorskie oraz majątkowe są jasno określone i nie są przekazywane użytkownikowi. Nie może on kopiować nabytego produktu, używać na więk-

- szej, niż określona przez sprzedawcę stanowisk. Użytkownik otrzymuje program w postaci zrozumiałej dla komputera, nie wie jednak jak on działa.
- *ShareWare*. To forma zbliżona do licencji komercyjnej. Możliwe jest jednak wykorzystywanie programu przez określony okres lub określoną liczbę uruchomień. Jest to czas na testowanie i decyzję o zakupie. Po zakończeniu okresu próbnego należy kupić program lub zrezygnować z niego całkowicie.
  - *FreeWare*. Autorzy udostępniają program za darmo, jednakże zachowują sobie kody źródłowe, prawa do modyfikacji, prawa do zmiany licencji. Nierzadko nowe programy rozprowadzane są jako darmowe, a gdy osiągną odpowiedni stopień rozwoju i popularności, autorzy każą za nie płacić.
  - *Wolne Oprogramowanie, np. GNU GPL*. Autorzy tworzą oprogramowanie i udostępniają go razem z kodem źródłowym. Mogą udostępnić za darmo albo za opłatą. Jednakże zawsze razem z programem muszą dostarczyć kod źródłowy. Nabywca może sprawdzić działanie programu, może modyfikować go na własne potrzeby. Warunkiem jest jednak to, że jeśli zmodyfikuje coś w programie lub wykorzysta jego fragmenty w swoim własnym produkcie, zobowiązany jest rozpowszechniać go także na licencji GNU GPL, czyli również z kodem źródłowym. Dzięki temu wszystkie ulepszenia i dodatki są dalej dostępne i kolejne osoby mogą je rozwijać. Autorzy (wszyscy) zachowują swoje prawa. Nikt nie może zabrać kodu na licencji GNU GPL i powiedzieć, że to jego dzieło lub zmienić tą licencję na komercyjną. Dlatego właśnie tak wielu programistów korzysta z tej licencji. Dają coś światu, a dzięki licencji Wolnego Oprogramowania wiedzą, że kolejne osoby ulepszające ich pracę, będą zobowiązane jej wyniki również udostępnić.
  - *Public domain* (dobro publiczne). To rzadka i nietypowa licencja w stylu *Róbta co chceta*. Autorzy tworzą coś i oddają to bez zobowiązań, bez jakichkolwiek praw. To nie ma nic wspólnego z Wolnym Oprogramowaniem, gdyż ktoś może zabrać kod źródłowy dostępny na licencji *public domain*, zmodyfikować go i sprzedawać jako komercyjny.

Mitem, jaki czasami się słyszy, jest stwierdzenie, że Wolne Oprogramowanie nie może być wykorzystywane komercyjnie. Nic bardziej błędnego! Produkty na licencji GNU GPL oraz pokrewnych, mogą być wykorzystywane zarówno prywatnie, jak i w przedsiębiorstwach. Możliwe jest również ich sprzedawanie. Często zdarza się, że klient zamawia oprogramowanie na swoje konkretne potrzeby. Można wtedy skorzystać z dostępnego na licencji GNU GPL oprogramowania, zmodyfikować i dostosować je do określonych potrzeb, a następnie pobrać za to pieniądze. W modelu biznesowym Wolnego Oprogramowania zarabia się na przeróbkach, dostosowaniu do potrzeb klienta oraz na usługach.