

# LINUX+

NAJWIĘKSZY EUROPEJSKI MAGAZYN O LINUJSIE

NR 12/2010 (162)

**BUDOWANIE APLIKACJI INTERNETOWYCH W APTANA STUDIO**

**CERTYFIKATY AIRTRUST**

**GWARANCJA AUTENTYCZNOŚCI TWOICH APLIKACJI**

**BIBLIOTEKA QT**

**PROFESJONALNY FRAMEWORK DLA KAŻDEGO**

**STATYCZNE SYSTEMY CMS**

**ZA SOFTWARE JAK ZA WODĘ**

**LINUX KERNEL:**

**CZĘŚĆ PIĄTA (5/7):**

**STEROWNIKI URZĄDZEŃ**

**PODRÓŻ DO WNĘTRZA SYSTEMU**

# Praca na wysokim poziomie



Serwis rekrutacyjny dla menedżerów i specjalistów  
[www.stepstone.pl](http://www.stepstone.pl)

# **Software Developer's JOURNAL**

*new ideas & solutions for professional programmers*

NR 12 GRUDZIEŃ 2010 (192)

## **zrób to w chmurze**

### **Amazon EC2 i Google App Engine**

BIBLIOTEKA MIESIĄCA

STL W PRAKTYCE: B USUWANIE ELEMENTÓW  
EFEKTYWNE USUWANIE ELEMENTÓW

**Pobierz nowy numer SDJ!**  
**>> KLIKNIJ <<**

#### **Java – Problemy**

Narzędzia, strategie oraz sposoby rozwiązywania problemów wydajnościowych z maszyną wirtualną Java

#### **Asynchroniczna obsługa urządzeń wejścia – wyjścia**

Biblioteka boost::asio

#### **Formalne Inspekcje**

Sprawdzony sposób na poprawę jakości kodu

#### **Kariera**

Niemiecki rynek czeka na polskich specjalistów IT

PARTNERZY WYDANIA

**accenture**

*High performance. Delivered.*

**LUXOFT**  
Engineering Business Performance

## SPIS TREŚCI

### JĄDRO LINUKSA

#### 6 Linux Kernel: Podróż do wnętrza systemu Część piąta (5/7): Sterowniki urządzeń

Rafał Kułaga

W poprzedniej części cyklu przedstawione zostały mechanizmy jądra kluczowe dla działania systemu operacyjnego: obsługa zadań, wypołania systemowe oraz system plików procfs. Kolejnym obszarem, w którym umiejętność programowania w przestrzeni jądra jest niezbędna jest tworzenie sterowników urządzeń. Z punktu widzenia przeciętnego użytkownika jest to zadanie niezwykle skomplikowane – w trakcie lektury tego artykułu przekonasz się jednak, iż sprowadza się ono do wykorzystania odpowiednich interfejsów jądra, przeznaczonych do obsługi danego urządzenia. Zapraszam do lektury!

### ROZWIĄZANIA

#### 16 Budowanie aplikacji internetowych w Aptana Studio

Paweł Wolniewicz

Aptana sprawdzi się szczególnie dobrze, jeśli HTML, CSS, DOM, JavaScript, PHP i/lub Ruby on Rails to dla nas codzienność. Środowisko to oferuje również atrakcyjny dodatek – możliwość integracji z Eclipse. Oczywiście Aptana Studio działa również jako w pełni samodzielna aplikacja. Zaletę stanowi wieloplatformowość. Napisana w języku Java Aptana zadziała w każdym spośród popularnych systemów operacyjnych.

#### 24 Stacyczne systemy CMS

Kacper Pluta

Jedną z niezaprzecjalnie mniej licznych grup systemów zarządzania treścią jest grupa systemów generujących statyczną treść. Dzięki temu, że pliki generowane są statycznie, możemy używać tego typu CMS-ów na naszej domowej maszynie, a dopiero później synchronizować zmiany. Takie podejście pozwala wyeliminować wiele wad dynamicznych systemów treści

### BEZPIECZEŃSTWO

#### 28 Certyfikaty AIRTrust. Gwarancja autentyczności Twoich aplikacji

Sebastian Fyda

Podpisując swoje aplikacje certyfikatem Adobe Air, udowadniasz swoją tożsamość użytkownikom, co

W kolejnym  
numerze ukaże się  
wywiad z Linusem  
Torvaldsem

przekłada się bezpośrednio na wzrost zaufania do Twojego produktu i Twojej firmy, jednocześnie pokazujesz użytkownikom, że Twój oprogramowanie można bezpiecznie zainstalować i używać.

## WARSZTATY

### 32 Biblioteka Qt – profesjonalny framework dla każdego

*Rafał Kułaga*

Istnieje wiele różnic pomiędzy amatorskim programowaniem a profesjonalnym tworzeniem systemów informatycznych. Oprócz oczywistych, takich jak ogromna różnica w złożoności kodu i liczby osób zaangażowanych w projekt, jedną z najważniejszych jest zastosowanie odpowiednich bibliotek i frameworków programistycznych. Dostarczą one klas do obsługi tak podstawowych technologii, jak komunikacja w protokole TCP/IP, dokumentów XML, graficznych interfejsów użytkownika oraz programowania zgodnego ze wzorcem model-widok-kontroler (MVC – Model-View-Controller). Jedną z najlepszych i najszerzej stosowanych

Miesięcznik **Linux+** (12 numerów w roku) jest wydawany przez Software Press Sp. z o.o. SK

**Redaktor naczelny i korekta:**  
Tomasz Łopuszański  
[tomasz.lopuszanski@software.com.pl](mailto:tomasz.lopuszanski@software.com.pl)

**Skład i łamanie:**  
Tomasz Kostro  
[www.studiopoligraficzne.com](http://www.studiopoligraficzne.com)

**Kierownik produkcji:**  
Andrzej Kuca  
[andrzej.kuca@software.com.pl](mailto:andrzej.kuca@software.com.pl)

**Stali współpracownicy:**  
Andrzej Jankowski, Roger Zacharczyk,  
Leszek Konka, Piotr Brzózka,  
Robert Romanik, Sławomir Iwanek

**Adres korespondencyjny:**  
Software Press Sp. z o.o. SK,  
ul. Bokserska 1, 02-682 Warszawa,  
Polska  
tel. +48 22 427 36 91,  
fax +48 22 224 24 59  
[www.sdjournal.org](http://www.sdjournal.org)  
[cooperation@software.com.pl](mailto:cooperation@software.com.pl)

**Dział reklamy:** [adv@software.com.pl](mailto:adv@software.com.pl)

Redakcja dokłada wszelkich starań, by publikowane w piśmie i na towarzyszących mu nośnikach informacje i programy były poprawne, jednakże nie bierze odpowiedzialności za efekty wykorzystania ich; nie gwarantuje także poprawnego działania programów shareware, freeware i public domain. Wszystkie znaki firmowe zawarte w piśmie są własnością odpowiednich firm. Zostały użyte wyłącznie w celach informacyjnych.

**Osoby zainteresowane współpracą prosimy o kontakt:**  
[cooperation@software.com.pl](mailto:cooperation@software.com.pl)

bibliotek programistycznych jest Qt. Ten krótki artykuł stanowi pilota nowego cyklu poświęconego programowaniu z wykorzystaniem tej biblioteki.

## RAPORT

### 36 Za software jak za wodę

*Magdalena Kołodziej*

Zmiany na rynku spowodowane kryzysem gospodarczym postawiły nowe wyzwania przed sektorem IT. Firmy potrzebują nowych rozwiązań, by móc maksymalizować zyski z projektów, obniżając jednocześnie ryzyko. Era długoterminowych kontraktów powoli miną, dzisiaj przedsiębiorcy za kadrę, sprzęt i oprogramowanie IT chcą płacić tak, jak za wodę - według zużycia.

## PINGWIN NA LUZIE

### 38 Jak przekonać kobietę do Linuxa

*Beata Rokicka*

Reklama



EBITDA Sp. z o.o.  
ul. Szarych Szeregów 27  
60-462 Poznań  
[biuro@serwer-firmowy.pl](mailto:biuro@serwer-firmowy.pl)

**SERWER-FIRMOWY.PL**

ZAWSZE DOSTĘPNY.... ZAWSZE BEZPIECZNY...  
ZAWSZE GOTOWY DO PRACY...

Nie wymaga Twojego serwisu, nie nawala w nim dysk, karta pamięci, nie utraça danych i nie zależy go sąsiad...

**Serwer Firmowy** to prawdziwy **Ciązarowiec** w Twojej firmie. Nie ma zadań, których nie udźwignie. Pracy na nim biuro, księgowość, mogą działać strony www czy serwer poczty umożliwiający pracę zdalną w grupie.

A kosztuje mniej niż kawa do Biura

infolinia  
**0801 00 30 37**

# Linux Kernel: Podróż do wnętrza systemu

## Część piąta (5/7): Sterowniki urządzeń

W poprzedniej części cyklu przedstawione zostały mechanizmy jądra kluczowe dla działania systemu operacyjnego: obsługa zadań, wywołania systemowe oraz system plików procfs. Kolejnym obszarem, w którym umiejętność programowania w przestrzeni jądra jest niezbędna jest tworzenie sterowników urządzeń. Z punktu widzenia przeciętnego użytkownika jest to zadanie niezwykle skomplikowane – w trakcie lektury tego artykułu przekonasz się jednak, iż sprowadza się ono do wykorzystania odpowiednich interfejsów jądra, przeznaczonych do obsługi danego urządzenia. Zapraszam do lektury!

Dzisiejsze komputery składają się z dziesiątek współpracujących ze sobą urządzeń. Użytkownicy często nie zdają sobie sprawy z tej złożoności, traktując komputer jako swego rodzaju *czarną skrzynkę*, która posiada pewne wejścia oraz wyjścia. Jest to oczywiście w pełni uzasadnione – zasada działania poszczególnych komponentów dawno przestała być potrzebna w celu efektywnego wykorzystania możliwości sprzętu do pracy oraz rozrywki.

Systemy operacyjne, w tym te oparte na jądrze Linux, ukrywają przed użytkownikiem złożoność wykorzystywanej przez niego sprzętu, dostarczając wygodnego środowiska do uruchamiania rozmaitych aplikacji. Również instalacja nowych urządzeń odbywa się w sposób przyjazny dla użytkownika. Jest to możliwe dzięki zastosowaniu technologii takich jak *Plug and Play*, pozwalających na automatyczne rozpoznanie urządzenia oraz przydzielenie mu zasobów systemowych, takich jak np. identyfikatory przerwań IRQ oraz umożliwiających podłączanie urządzeń w trakcie pracy komputera (*Hot Plug*).

Programowanie sterowników jest jednak czynnością, która wymaga głębszej wiedzy na temat architektury systemu operacyjnego oraz dostępu do pełnej dokumentacji urządzenia, dla którego sterownik jest tworzony. Oczywiście zdarzają się przypadki, iż sterowniki są tworzone metodami inżynierii wstępnej przez osoby nie posiadające dostępu do dokumentacji, jednak jest

to zadanie bardzo trudne w przypadku urządzeń o złożonej funkcjonalności. Wykorzystanie inżynierii wstępnej może wiązać się również ze złamaniem warunków umowy licencyjnej, która bardzo często zabrania stosowania tego typu praktyk.

W artykule tym zapoznasz się z podstawowymi informacjami na temat tworzenia sterowników dla jądra Linux. Jako, że jest to temat niezwykle rozległy oraz wymagający solidnej znajomości magistrali i interfejsów, z których korzysta urządzenie, dla którego piszemy sterownik, w artykule prezentowane będą jedynie ograniczone przykłady. Nawet prosty sterownik składa się z bardzo wielu linijek kodu, z których większość byłaby dla Czytelników niezrozumiała z powodu nieznajomości sprzętu. W artykule skupimy się na opisie typów urządzeń obsługiwanych przez system Linux, a w szczególności na urządzeniach znakowych (*Character devices*) oraz zapoznamy się z mechanizmami obsługi przerwań.

### Sterowniki urządzeń – charakterystyka

Sterowniki urządzeń są modułami jądra, które po przez wykorzystanie odpowiednich interfejsów systemu umożliwiają konfigurację i wykorzystanie sprzętu podłączonego do systemu. Każdy sterownik może oczywiście zostać włączony do statycznej części jądra, jednak w praktyce rozwiązanie takie stosuje się jedynie w przypadku sterowników podstawowych urządzeń

wejścia/wyjścia oraz urządzeń systemowych, takich jak np. zegar czasu rzeczywistego.

Oprócz podstawowej funkcjonalności, jaką jest umożliwienie dostępu do urządzeń przez programy przestrzeni użytkownika, sterowniki umożliwiają również abstrakcję pewnych aspektów pracy urządzeń. Dla przykładu: różne modele drukarek wymagają odmiennych sterowników, wszystkie jednak pozwalały na ich wykorzystanie w podobny sposób, co jest możliwe dzięki użyciu odpowiednich interfejsów systemu. Odbywa się to w sposób podobny do przedstawionego już systemu plików procfs.

## Urządzenia w Linuksie

Każdy użytkownik systemu Linux wie, iż pliki reprezentujące urządzenia zarejestrowane w systemie znajdują się w katalogu /dev. Nie wszystkie reprezentują jednak rzeczywiste urządzenia systemowe – przykładem urządzenia wirtualnego może być /dev/random, będący generatorem liczb losowych. Przykładem pliku urządzenia jest /dev/ttys0 reprezentujący port szeregowy lub /dev/fb reprezentujący bufor ramki graficznej.

W systemie Linux (oraz w większości systemów operacyjnych) istnieje podział na dwa podstawowe typy urządzeń: urządzenia znakowe oraz blokowe. Urządzenia znakowe charakteryzują się komunikacją realizowaną za pomocą strumieni znaków – jednocześnie przesyłany lub odbierany jest jeden znak. W transmisji realizowanej w ten sposób zazwyczaj nie korzysta się z buforów dla odbieranych i przesyłanych danych. Przykładem sprzętu tego typu może być klawiatura, modem lub dowolne urządzenie podłączone do portu szeregowego.

Urządzenia blokowe, do których należą dyski twardy oraz pamięci USB, działają w odmienny sposób. Transmisja danych bajt po bajcie byłaby w nich przypadku byłaby mało wydajna. Z tego względu, w przypadku urządzeń blokowych, przesyłane i odbierane dane są buforowane. Transmisja ma miejsce dopiero po zapełnieniu bufora.

W celu sprawdzenia typu urządzeń znajdujących się w katalogu /dev, należy skorzystać z polecenia ls /dev -o (Rysunek 1). Linie rozpoczętające się znakiem c wskazują na pliki reprezentujące urządzenia znakowe, zaś rozpoczynające się znakiem b – urządzenia blokowe. Warto również zwrócić uwagę na linie rozpoczynające się znakiem l, które wskazują na łącza symboliczne.

## Urządzenia znakowe

Urządzenia znakowe należą do najczęściej wykorzystywanych urządzeń w systemie Linux. Charakteryzują się one nieskomplikowanym interfejsem, który zostanie omówiony w tym paragrafie.

W artykule podane już zostały przykłady urządzeń znakowych – nie są to oczywiście wszystkie urządzenia tego typu zarejestrowane w systemie. Nie każde z urządzeń znakowych musi być również powiązane z fizycznym sprzętem. We wszystkich nowoczesnych systemach operacyjnych pojęcie urządzenia jest znacznie bardziej abstrakcyjne i odnosi się również do urządzeń wirtualnych.

## Interfejs urządzeń znakowych

Urządzenia znakowe w systemie Linux identyfikowane są przez dwie liczby całkowite: identyfikator główny (*major*) oraz poboczny (*minor*). Mogą one zostać odczytane przy użyciu polecenia ls /dev -o – znajdują się one w dwóch kolumnach oddzielonych przecinkiem. Numer główny służy do określenia sterownika urządzenia, zaś numer poboczny wykorzystywany jest do identyfikacji urządzeń wewnętrz sterownika.

Należy pamiętać, iż numery główne oraz poboczne nie są przydzielane tylko urządzeniom znakowym – występują również w przypadku urządzeń blokowych.

Każde z urządzeń znakowych zarejestrowanych w systemie posiada strukturę typu file\_operations. Służy ona do przechowywania wielu zmiennych związanych z obsługą urządzenia znakowego, w tym wskaźniki do funkcji obsługujących takie akcje jak np. odczyt lub zapis do urządzenia. Najważniejsze z pól tej struktury to:

- read – wskaźnik do funkcji obsługującej odczyt z urządzenia;

```
Terminal
Plik Edycja Widok Terminal Karty Pomoc
debian:/dev# ls -o
razem 0
crw-rw---- 1 root 14, 12 paź 8 16:02 adsp
crw-rw---- 1 root 14, 4 paź 8 16:02 audio
drwxr-xr-x 3 root 60 paź 8 16:02 bus
lrwxrwxrwx 1 root 3 paź 8 16:02 cdrom -> hdc
crw----- 1 root 5, 1 paź 8 14:02 console
lrwxrwxrwx 1 root 11 paź 8 16:02 core -> /proc/kcore
crw-rw---- 1 root 10, 63 paź 8 16:02 cpu_dma_latency
drwxr-xr-x 5 root 100 paź 8 16:02 disk
crw-rw---- 1 root 14, 3 paź 8 16:02 dsp
lrwxrwxrwx 1 root 3 paź 8 16:02 dvd -> hdc
lrwxrwxrwx 1 root 13 paź 8 16:02 fd -> /proc/self/fd
brw-rw---- 1 root 2, 0 paź 8 16:02 fd0
crw-rw-rw- 1 root 1, 7 paź 8 16:02 full
brw-rw---- 1 root 3, 0 paź 8 16:02 hda
brw-rw---- 1 root 3, 1 paź 8 16:02 hda1
brw-rw---- 1 root 3, 2 paź 8 21:52 hda2
brw-rw---- 1 root 3, 5 paź 8 16:02 hda5
brw-rw---- 1 root 22, 0 paź 8 16:02 hdc
crw-rw---- 1 root 10, 228 paź 8 16:02 hpet
prw----- 1 root 0 paź 8 16:02 initctl
drwxr-xr-x 3 root 200 paź 8 16:02 input
crw-r----- 1 root 1, 2 paź 8 16:02 kmem
```

Rysunek 1. Wynik działania polecenia ls /dev -o

**Listing 1a.** Przykład prostego wirtualnego urządzenia znakowego

```

#include <linux/kernel.h>
#include <linux/module.h>
#include <linux/fs.h>
#include <asm/uaccess.h>

#define DEV_NAME "character_device"

MODULE_LICENSE("GPL");
MODULE_AUTHOR("R. Kulaga");
MODULE_DESCRIPTION("Przyklad prostego wirtualnego
                   urzadzenia znakowego");

static int dev_open(struct inode *, struct file *);
static int dev_release(struct inode *, struct file *);
static ssize_t dev_read(struct file *, char *, size_t,
                       loff_t *);

static int major; /* liczba glowna urzadzenia */
static int opened; /* licznik otworzen */
static char msg[128]; /* wiadomosc do wyswietlenia */
static char *msg_ptr; /* wskaznik do wiadomosci */

/* struktura przechowujaca wskazniki do funkcji
   obslugujacych */
static struct file_operations fops = {
    .read = dev_read,
    .open = dev_open,
    .release = dev_release
};

int init_module()
{
    /* zarejestrowanie urzadzenia w jadrze */
    major = register_chrdev(0, DEV_NAME, &fops);

    /* rejestracja urzadzenia nie powiodla sie */
    if(major < 0)
    {
        printk(KERN_ALERT "Rejestracja urzadzenia nie
                      powiodla sie. Kod bledu: %d\n",
                      major);
        return major;
    }

    printk(KERN_INFO "Rejestracja urzadzenia zakonczona
                  sukcesem. Numer glowny: %d\n",
                  major);

    return 0;
}

void cleanup_module()
{
    /* wyrejestrowanie urzadzenia znakowego */
    unregister_chrdev(major, DEV_NAME);
}

static int dev_open(struct inode *inode, struct file *file)
{
    /* licznik odczytow z urzadzenia */
    static int counter = 0;

    /* inkrementacja licznika procesow korzystajacych z
       modulu*/
    try_module_get(THIS_MODULE);

    /* jezeli urzadzenie jest juz otwarcone */
    if(opened)
    {
        return -EBUSY;
    }

    /* inkrementacja licznik otworzen */
    opened++;

    /* przygotowanie tekstu wiadomosci */
    sprintf(msg, "Wartosc licznika otworzen: %d\n",
            counter++);

    /* ustawienie wskaznika wiadomosci */
    msg_ptr = msg;

    /* return 0;
    */

    static int dev_release(struct inode *inode, struct file
                          *file)
    {
        /* dekrementacja licznika otworzen */
        opened--;

        /* dekrementacja licznika procesow korzystajacych z
           modulu */
        module_put(THIS_MODULE);

        return 0;
    }

    static ssize_t dev_read(struct file *filp,
                           char *buffer,
                           size_t length,
                           loff_t *offset)
    {
        /* licznik odczytanych bajtow */
    }
}

```

**Listing 1b.** Przykład prostego wirtualnego urządzenia znakowego

```

int bytes_read = 0;

/* koniec wiadomości oznaczony jest znakiem NULL*/
if(*msg_ptr == 0)
{
    return 0;
}

/* skopiowanie danych do bufora */
while(length && *msg_ptr)
{
    /* skopiowanie danych do przestrzeni uzytkownika */
    /* */
    put_user(*(msg_ptr++), buffer++);

    length--;
    bytes_read++;
}
}

return bytes_read;
}

```

- `write` – wskaźnik do funkcji obsługującej zapis danych do urządzenia;
- `open` – wskaźnik do funkcji obsługującej otwarcie urządzenia;
- `release` – wskaźnik do funkcji obsługującej zwolnienie urządzenia;
- `owner` – wskaźnik do struktury typu `module`, wskażujący na moduł będący właścicielem urządzenia znakowego.

**Rejestracja urządzeń znakowych**

Utworzenie nowego urządzenia znakowego odbywa się przy użyciu funkcji `register_chrdev()` o prototypie

```
int register_chrdev(unsigned int major,
                    const char *name,
                    struct file_operations *fops)
```

Funkcja przyjmuje trzy argumenty i zwraca wartość typu `int`. Argument `major` określa żądaną liczbę główną urządzenia zaś `name` – nazwę urządzenia w katalogu `/dev`. Ostatni z argumentów – `fops` – jest wskaźnikiem do omówionej już struktury typu `file_operations`, zawierającej wskaźniki do funkcji obsługujących akcje na urządzeniu.

Szczególną uwagę należy poświęcić przekazywanej liczbie głównej. Musi ona być bowiem unikatowa w obrębie całego systemu. Z tego względu, dobrym rozwiązaniem jest przekazanie `0` jako wartości tego argumentu – spowoduje to dynamiczne przydzielenie wolnego numeru głównego, który będzie w takim przypadku wartością zwracaną przez funkcję. Zwrócenie ujemnej wartości świadczy o wystąpieniu błędu w trakcie rejestracji urządzenia.

**Usunięcie urządzenia znakowego**

Przed odłączeniem od jądra modułu, który zarejestrował urządzenia znakowe, powinno zawsze nastąpić ich wyrejestrowanie. Pominięcie tego kroku może skutkować poważnymi błędami o trudnych do przewidzenia konsekwencjach.

Wyrejestrowanie urządzenia znakowego odbywa się przy użyciu funkcji `unregister_chrdev()` o prototypie

```
void unregister_chrdev(unsigned int major, const char
                      *name)
```

Funkcja przyjmuje dwa argumenty. Argument `major` określa numer główny wyrejestrowanego urządzenia, zaś `name` – jego nazwę.

**Przykładowy moduł  
– wirtualne urządzenie znakowe**

Przykładowy sterownik wirtualnego urządzenia znakowego zawarty został na Listingu 1. Większość z wykorzystanych w nim funkcji została już omówiona w tej oraz w poprzednich częściach cyklu. W razie wątpliwości co do ich znaczenia, polecam odwołać się do ich treści.

Krótko prześledźmy działanie modułu: w funkcji `init_module()` rejestrowane jest nowe urządzenie znakowe o nazwie określonej przez `DEV_NAME` oraz dynamicznie przydzielonym numerze głównym. Do funkcji `register_chrdev()` przekazany zostaje również wskaźnik do struktury typu `file_operations`, zawierającej wskaźniki do funkcji obsługujących otwarcie, odczyt oraz zamknięcie pliku. Przedstawiony moduł nie umożliwia zapisu do urządzenia, toteż nie implementujemy funkcji `write`.

Działanie funkcji `dev_read()`, `dev_open()` oraz `dev_release()` zostało opisane przy użyciu komentarzy. Szczególną uwagę warto zwrócić na funkcje `try_module_get()` oraz `module_put()`, służące do modyfikacji wartości licznika procesów korzystających z modułu.

W funkcji `cleanup_module()` urządzenie znakowe zostaje wyrejestrowane w sposób opisany w poprzednim paragrafie.

Do komplikacji modułu z Listingu 1 można wykorzystać plik Makefile zawarty na Listingu 2. Po komplikacji i dołączeniu modułu konieczne jest jeszcze utworzenie odpowiedniego pliku urządzenia. Można tego dokonać

przy pomocy polecenia `mknod /dev/DEV_NAME c major 0`, gdzie `DEV_NAME` – nazwa urządzenia, `major` – jego liczba główna.

Po wykonaniu opisanych powyżej czynności można już sprawdzić działanie modułu korzystając z polecenia `cat /dev/DEV_NAME`. Po odłączeniu modułu plik urządzenia nie zostanie usunięty z katalogu `/dev`; należy zrobić to własnoręcznie.

## Urządzenia blokowe

Urządzenia blokowe charakteryzują się znacznie większą funkcjonalnością i wydajnością niż urządzenia znakowe. Również interfejsy jądra, z których korzystają są bardziej skomplikowane i wymagają znajomości zagadnień związanych z alokacją oraz stronowaniem pamięci.

Omówieniem interfejsu jądra dla urządzeń blokowych zajmiemy się w następnej części cyku.

## Przerwania i ich obsługa w Linuksie

Jak dotąd w artykule zostały omówione przypadki, w których procesor nawiązuje komunikację z urządzeniem. Warto zastanowić się jednak, w jaki sposób sprzęt podłączony do komputera może informować procesor o przebiegu wykonania przydzielonych mu zadań. Problem ten staje się niebanalny, jeżeli uwzględnimy fakt, iż częstotliwość taktowania współczesnych, wielordzeniowych CPU sięga kilku gigaherców. Periodyczne odpytywanie urządzeń (*Polling*) stanowi bowiem marnotrawstwo mocy obliczeniowej oraz powoduje wiele innych problemów, które zostaną szerzej opisane w tym paragrafie.

Aby rozwiązać problemy pojawiające się przy wykorzystaniu odpytywania, sprzęt oraz jądro systemu operacyjnego wyposażane jest w obsługę przerwań.

## Działanie przerwań – przykład

Zgłoszenie przerwania odbywa się przy pomocy sygnału elektrycznego doprowadzanego do odpowiedniego wyprowadzenia procesora. Jako że w systemie istnieje wiele źródeł przerwań, wykorzystuje się tzw. kontrolery

przerwań – układy, których zadaniem jest multipleksowanie wielu linii przerwań do pojedynczej linii przerwania procesora.

Każde z przerwań posiada przypisany liczbowy identyfikator, zwany IRQ (*Interrupt Request*). Najważniejsze przerwania wykorzystywane w komputerach opartych na architekturze x86 zebrane zostały w tabeli *Najważniejsze przerwania w architekturze x86*. Obsługa przerwania przez jądro systemu operacyjnego omówiona zostanie na przykładzie przerwania 1 – przerwania klawiatury.

Przerwanie generowane jest przez klawiaturę po każdym naciśnięciu przycisku przez użytkownika. Nie będziemy tu dokładnie opisywać działania sterownika klawiatury – zainteresowanym polecam zapoznanie się z informacjami zawartymi w internecie. Po odebraniu żądania przerwania, procesor odczytuje jego numer, po czym jest ono obsługiwane przez jądro systemu przy pomocy odpowiedniej funkcji obsługującej, nazywanej ISR (*Interrupt Service Routine*).

## Problemy z odpytywaniem urządzeń (Polling)

Istnieje wiele powodów dla których odpytywanie urządzeń nie jest szeroko wykorzystywane. Najważniejsze z nich to:

- Narzut czasowy spowodowany odpytywaniem urządzeń – procesor odpytuje urządzenia, które nie muszą zgłaszać żądania obsługi. Jeżeli w systemie znajduje się duża liczba urządzeń, ich odpytywanie z określoną częstotliwością może zajmować stosunkowo duży procent czasu procesora. Dużym problemem jest również określenie optymalnej częstotliwości odpytywania urządzenia;
- Brak możliwości obsługi w oparciu o priorytety – w przypadku zastosowania przerwań możliwe jest określenie ich priorytetów. W przypadku zgłoszenia przerwania o większym priorytecie, aktualnie obsługiwane przerwanie o niższym priorytecie jest przerywane, przerwanie o wyższym priorytecie zostaje obsłużone, po czym kontynuowane jest wy-

**Tabela 1. Najważniejsze przerwania w architekturze x86**

Numer przerwania (heksadecymalnie)	Identyfikator IRQ	Urządzenie/znaczenie
08h	IRQ0	Zegar systemowy (timer)
09h	IRQ1	Klawiatura
0Bh	IRQ3	Porty szeregowe COM2/COM4
0Ch	IRQ4	Porty szeregowe COM1/COM3
0Dh	IRQ5	Karta dźwiękowa
0Fh	IRQ7	Port równoległy LPT
10h-6Fh	Nie dotyczy	Przerwania programowe
70h	IRQ8	Zegar czasu rzeczywistego (RTC)
74h	IRQ12	Mysz PS/2

- konywanie kodu ISR przerwania o niższym priorytecie. Przy zastosowaniu pollingu nie ma możliwości przerwania aktualnie obsługiwanej żądania;
- Czas odpowiedzi zależy od liczby urządzeń w systemie – odpytywanie każdego z urządzeń zajmuje czas procesora. W przypadku wzrostu liczby urządzeń wydłużeniu ulega sumaryczny czas ich obsługi. Zastosowanie przerwań pomaga zapobiec temu problemowi, bowiem czas reakcji na przerwanie (rozumiany jako czas pomiędzy zgłoszeniem przerwania a rozpoczęciem wykonywania funkcji obsługującej) jest stały.

Rozwiążanie, jakim jest zastosowanie odpytywania urządzeń, jest jak widać słabo skalowalne oraz powoduje wiele problemów. Zastosowanie tej metody obsługi urządzeń jest wykluczone w przypadku systemów czasu rzeczywistego – nie pozwala bowiem na definiowanie priorytetów oraz jest nieprzewidywalne pod względem czasowym.

### Funkcje obsługi przerwań i ich rejestracja

Aby umożliwić obsługę przerwania, konieczne jest za-rejestrowanie w jądrze odpowiedniej funkcji.

Deklaracja funkcji obsługującej przerwanie jest następująca

REKLAMA

```
static irqreturn_t interrupt_hndlr(int irq, void *dev)
```

Pierwszy argument funkcji `interrupt_hndlr` – `irq` – jest typu `int` i określa numer obsługiwanej przerwania. Argument `dev` jest identyfikatorem urządzenia podanego przy zarejestrowaniu przerwania w systemie. W przypadku gdy wiele urządzeń ma mieć możliwość korzystania z tej samej funkcji obsługującej przerwanie, wartość przekazana jako argument `dev` powinna być unikatowa.

Wartość zwracana przez funkcję jest typu `irqreturn_t`, zdefiniowanego jako

```
typedef irqreturn_t (*irq_handler_t)(int, void*)
```

Wartość zwracana może przyjmować dwie wartości specjalne:

- `IRQ_NONE` – zwracana w przypadku gdy ISR wykryje, iż urządzenie od którego pochodzi przerwanie nie jest urządzeniem określonym przez zmien-ną `dev`;
- `IRQ_HANDLED` – zwracana w przypadku gdy przerwanie zostało poprawnie obsłużone.

Wartość zwracana jest wartością typu `int`.



# masz dosyć?

**Branża IT nie musi być nudna**

**Rozpoczęliśmy rekrutację w Polsce**



Lubię to! [www.Facebook.com/Luxoft.Poland](http://www.Facebook.com/Luxoft.Poland)

Rejestracja przerwania w jądrze odbywa się przy użyciu funkcji `request_irq()` o prototypie

```
int request_irq(unsigned int irq,
    irq_handler_t handler,
    unsigned long flags,
    const char *name,
    void *dev)
```

Argument `irq` określa numer przerwania, dla którego określamy funkcję obsługującą, drugi argument – `handler` – zawiera wskaźnik do funkcji obsługującej przerwanie. Przy rejestracji funkcji obsługującej przerwanie można podać flagi wpływające na sposób jego obsługi. Najważniejsze z nich to:

- `IRQF_DISABLED` – ustawienie tej flagi powoduje zablokowanie wszystkich przerwań w momencie wykonywania rejestrowanej funkcji obsługującej. Flaga ta wykorzystywana jest w przypadku przerwań, których obsługa trwa krótko oraz ma krytyczne znaczenie dla pracy systemu;
- `IRQF_SHARED` – ustawienie tej flagi umożliwia współdzielenie pojedynczego przerwania przez wiele procedur obsługujących;
- `IRQF_SAMPLE_RANDOM` – ustawienie tej flagi powoduje włączenie informacji o momentach wystąpienia rejestrowanego przerwania do puli entropii, wykorzystywanej przez urządzenie `/dev/random` do generowania liczb losowych. Bardzo ważne jest, aby flaga ta była ustawiana jedynie w przypadku urządzeń, dla których przerwania występują w różnych odstępach, niemożliwych do określenia na podstawie jego typu oraz w dużym stopniu niezależnych np. od komunikacji sieciowej. Uaktywnienie tej flagi dla urządzeń generujących przerwania w stałych odstępach czasowych skutkowałoby pogorszeniem jakości generowanych liczb losowych, które mają kluczowe znacze-

nie dla algorytmów szyfrujących. Więcej informacji na temat generowania liczb losowych w jądrze Linux oraz budowie wirtualnego urządzenia `/dev/random`, Czytelnik znajdzie na stronach wymienionych w ramce *W Sieci*.

Czwarty argument – `name` – określa nazwę, pod którą przerwanie będzie widoczne w plikach `/proc/irq` oraz `/proc/interrupts`. Więcej informacji na temat zawartości pliku `/proc/interrupts` zawarto w dalszej części artykułu.

Argument `dev` wykorzystywany jest w przypadku dzielenia jednej linii przerwania przez wiele procedur obsługi. W przypadku gdy zaistnieje potrzeba zwolnienia jednej z nich, jest to możliwe poprzez przekazanie odpowiedniej wartości argumentu `dev`.

### Zwolnienie funkcji obsługi przerwania

Typową sytuację, w której konieczne jest zwolnienie funkcji obsługi przerwania, jest odłączenie modułu, który ją zarejestrował. Pominięcie tej czynności skutkować będzie w najlepszym przypadku komunikatem o błędzie; w przypadku mniej optymistycznym, gdy pod adresem, w którym w trakcie działania modułu znajdowała się procedura obsługi znajdzie się inny kod, następuje jego wykonanie, co może powodować trudne do przewidzenia skutki.

```
debian:/# cat /proc/interrupts
CPU0
 0:      228 XT-PIC-XT      timer
 1:     2141 XT-PIC-XT i8042
 2:        0 XT-PIC-XT cascade
 5:     2824 XT-PIC-XT Intel 82801AA-ICH
 6:        4 XT-PIC-XT floppy
 8:        2 XT-PIC-XT rtc0
 9:   44833 XT-PIC-XT acpi, vboxguest
10: 1124298 XT-PIC-XT ehci_hcd:usb2, eth0
11:        0 XT-PIC-XT ohci_hcd:usb1
12: 152828 XT-PIC-XT i8042
14:  99145 XT-PIC-XT ide0
15: 112952 XT-PIC-XT idel
NMI:        0 Non-maskable interrupts
LOC:  544066 Local timer interrupts
RES:        0 Rescheduling interrupts
CAL:        0 function call interrupts
TLB:        0 TLB shootdowns
TRM:        0 Thermal event interrupts
SPU:        0 Spurious interrupts
ERR:        0
MIS:        0
debian:/#
```

Rysunek 2. Lista przerwań zarejestrowanych w systemie zawarta w pliku `/proc/interrupts`

### Listing 2. Plik Makefile dla modułów z listingów

```
obj-m += chrdev.o
all:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules
clean:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
```

Zwolnienie funkcji obsługi przerwania odbywa się przy użyciu funkcji `free_irq()` o prototypie

```
void free_irq(unsigned int irq, void *dev)
```

W przypadku gdy podany identyfikator przerwania `irq` jest dzielony, argument `dev` określa funkcję obsługi, która powinna zostać zwolniona. Po usunięciu ostatniej funkcji obsługującej, przerwanie zostaje zablokowane.

### Sterowanie przerwaniami i liniami przerwań

Po przedstawieniu funkcji używanych w celu rejestracji oraz zwalniania procedur obsługi przerwań, warto zapoznać się z funkcjami oraz makrami wykorzystywany mi w celu sterowania przerwaniami.

W celu zablokowania wszystkich przerwań dla aktywnego procesora należy skorzystać z funkcji `local_irq_disable()`. Odblokowanie przerwań odbywa się przy użyciu funkcji `local_irq_enable()`.

Zapis aktualnego stanu przerwań w systemie można wykonać przez zastosowanie funkcji `local_irq_save()`, która jako argument przyjmuje zmienną typu `unsigned long`. Przywrócenie przerwań do poprzedniego stanu zapisanego w zmiennej odbywa się przy użyciu funkcji `local_irq_restore()`, z przekazaniem odpowiedniej zmiennej.

W przypadku gdy chcemy zablokować pojedynczą linię przerwania, należy skorzystać z funkcji `disable_irq()`, jako argument podając zmienną typu `unsigned int` stanowiącą jego identyfikator. Ponowne aktywowanie przerwania odbywa się przy użyciu funkcji `enable_irq()`, jako argument wywołania przekazując jego identyfikator.

W celu sprawdzenia aktualnego kontekstu (przerwanie/proces) należy skorzystać z makra `in_interrupt()` zwracającego wartość niezerową gdy znajdujemy się w kontekście przerwania, zaś 0 w przypadku kontekstu procesu. Sprawdzenie aktualnego kontekstu jest konieczne przed wykonaniem instrukcji powodujących blokowanie, których wykonanie jest niemożliwe w kontekście przerwania.

### Top halves i bottom halves

Funkcje obsługujące przerwania są zwykłymi funkcjami C o określonym prototypie. Oprócz tego, wymaga się, aby ich wykonywanie trwało możliwie jak najkrócej, ponieważ na ten czas zawieszane jest wykonywanie pozostałego kodu jądra oraz wszystkich procesów. Z tego powodu w jądrze Linux wprowadzono podział obsługi przerwania na górną (*Top half*) oraz dolną połowę (*Bottom half*).

Górna połowa jest procedurą obsługi przerwania (ISR), w której zazwyczaj dane odbierane są od urządzeni.

REKLAMA



# Szukamy najlepszych informatyków

## Rozpoczęliśmy rekrutację w Polsce



Lubię to! [www.facebook.com/Luxoft.Poland](http://www.facebook.com/Luxoft.Poland)

dzenia, zaś sam sprzęt przygotowywany jest do dalszego działania (opróżnienie buforów, ustawienie odpowiednich rejestrów). Dolną połowę stanowi kod odpowiedzialny za obróbkę oraz wykorzystanie odebranych danych.

Obsługa przerwania odbywa się w kontekście zwanym kontekstem przerwania lub kontekstem atomowym. W kontekście tym zabronione jest wywoływanie funkcji które mogą skutkować zablokowaniem.

Przykładem podziału zadań obsługi przerwania na górną oraz dolną połowę może być obsługa przerwania od interfejsu sieciowego. Gdy z sieci odebrane zostaną ramki przeznaczone dla naszego komputera, karta zgłasza przerwanie, które zostaje obsłużone przez system przy pomocy odpowiedniej funkcji ISR, w której dane zostają skopiowane do pamięci, zaś interfejs przygotowany do dalszej pracy, po czym kończy się właściwa obsługa przerwania. W dolnej połowie odebrane dane są przetwarzane przez stos TCP/IP systemu, w którym po odłączeniu nagłówków poszczególnych protokołów, zostają przekazane odpowiedniej aplikacji przestrzeni użytkownika w postaci komunikatu.

## Plik /proc/interrupts

Informacje o przerwaniach zarejestrowanych w systemie można uzyskać wydając polecenie `cat /proc/interrupts`. Przykładowy wynik działania tego polecenia przedstawiony został na Rysunku 2. Poszczególne linie pliku odpowiadają przerwaniom zarejestrowanym w systemie, na końcu znajdują się zbiorcze informacje na temat obsługiwanych przerwań.

Linie reprezentujące poszczególne źródła przerwań składają się z kilku kolumn. W pierwszej z nich zawarto numer przerwania (IRQ), kolejne kolumny przedstawiają liczbę zgłoszeń danego przerwania dla wszystkich procesorów w systemie (w przypadku systemów z jednym procesorem będzie to oczywiście pojedyncza kolumna). Kolejna kolumna zawiera nazwę kontrolera przerwań obsługującego daną linię. Ostatnia kolumna zawiera nazwę urządzenia korzystającego z danego

przerwania – odpowiada ona argumentowi `name` przekazanemu w wywołaniu funkcji `request_irq()`.

W przykładzie przedstawionym na Rysunku 2 możemy odczytać, iż dla przerwania pochodzącego od zegara czasu rzeczywistego (`rtc0`) o IRQ wynoszącym 8 obsłużono dwa żądania, zaś kontrolerem jest XT-PIC-XT.

## Podsumowanie

W powyższym artykule opisane zostały podstawowe zagadnienia związane z tworzeniem sterowników i obsługą sprzętu w systemie Linux. Oprócz przedstawienia krok po kroku procesu tworzenia modułu stanowiącego sterownik wirtualnego urządzenia znakowego, opisana została obsługa przerwań. Artykuł ten należy traktować jako wstęp do tworzenia sterowników dla systemu Linux – szczegółowe omówienie tego tematu jest bowiem niemożliwe w przyjętej formie cyklu.

Czytelnikom szczególnie zainteresowanym tworzeniem sterowników dla systemu Linux polecam zapoznanie się z informacjami zawartymi na stronach wymienionych w ramce *W Sieci* oraz publikacjami książkowymi wymienionymi w ramce *Polecana literatura*. Pozycje na niej zawarte, a w szczególności te ściśle poświęcone tworzeniu sterowników z pewnością pozwolą Czytelnikowi na szersze poznanie tematu.

W następnej części cyklu zajmiemy się zagadnieniami związanymi z zarządzaniem pamięcią oraz wirtualnym systemem plików VFS (Virtual File System). Czytelnik poznaje mechanizmy związane z alokacją pamięci przez jądro, jej stronicowaniem oraz interfejsami umożliwiającymi współpracę różnych systemów plików – VFS stanowi bowiem wspólny interfejs pozwalający na dostęp do danych zawartych na nośnikach w sposób niezależny od wykorzystywanej na nim systemu plików. Do usłyszenia!

## RAFAŁ KUŁAGA

*Autor interesuje się bezpieczeństwem systemów informatycznych, programowaniem, elektroniką, muzyką rockową, architekturą mikroprocesorów oraz zastosowaniem Linuksa w systemach wbudowanych.*

*Kontakt z autorem: rl.kulaga@gmail.com*

## Polecana literatura

- Robert Love, *Linux Kernel Development*, wyd. Addison-Wesley;
- Sreekrishnan Venkateswaran, *Essential Linux Device Drivers*, wyd. Prentice Hall;
- Jonathan Corbet, Greg Kroah-Hartman, Alessandro Rubini, *Linux Device Drivers*, wyd. O'Reilly;

## W Sieci

- Strona główna jądra Linux - <http://www.kernel.org/>
- Strona KernelTrap - <http://kerneltrap.org/>
- Opis sterownika klawiatury PC (PS/2 oraz AT) - [http://www.freescale.com/files/microcontrollers/doc/app\\_note/AN1723.pdf](http://www.freescale.com/files/microcontrollers/doc/app_note/AN1723.pdf)



## pureAdvantage CHECK POINT ATC



Centrum Szkoleniowe Sp. z o.o. zaprasza serdecznie wszystkich zainteresowanych na listopadowe i grudniowe szkolenia firmy Check Point Software Ltd. z cyku

### **CCSA R70 i CCSE R70**

#### **Check Point Security Administrator R70 (CCSA R70)**

(czas trwania szkolenia: 40 godzin)

Kurs ten jest podstawowym szkoleniem przygotowującym uczestników do sztuki tworzenia polityk bezpieczeństwa z wykorzystaniem zapory ogniowej (firewall) tworzonej za pomocą najnowszej wersji oprogramowania NGX R70. Przedmiotem kursu jest System Zarządzającym Bezpieczeństwem Check Point oraz Zapora Ogniowa Check Point. Kurs dostarcza wiedzę o podstawowych konceptach i możliwościach operacyjnych takich składników (Blades) systemu jak: Zapora, VPN IPSEC, IPS, System Zarządzania Politykami, Logowanie i Status, Monitorowanie, Filtrowanie URL-i, Antywirus, Anty-Spam, Bezpieczeństwo Email. Podczas szkolenia uczestnicy wielokrotnie konfigurują polityki bezpieczeństwa, chronią segmenty sieciowe przed atakami i włamaniemi oraz uczą się zarządzania i monitorowania sieci zabezpieczonych.

#### **Check Point Security Expert R70 (CCSE R70)**

(czas trwania szkolenia: 40 godzin)

Kurs jest kontynuacją zaawansowanej nauki tworzenia polityk bezpieczeństwa z wykorzystaniem zapory ogniowej (firewall) firmy Check Point.

Kurs tłumaczy dalsze nowe zagadnienia z zakresu nauki produktu Check Point FireWall-1 R70 jak: Webowy Portal Zarządzający, SmartWorkflow, SmartProvisioning, a także pokazuje jak produkt zarządzający jest zintegrowany z Connectrą, czyli realizatorem wirułalnych sieci prywatnych SSL VPN. Nowością szkolenia jest również akceleracja SecureXL oraz jej wersja wielordzeniowa CoreXL. W kursie pojawia się również tematyka nadmiarowości rozwiązań zarządzających i klastra Zapór Ogniowych (ClusterXL) - tutaj między innymi zwrócono uwagę na zastosowanie protokołu VRRP na urządzeniach Nokia. W dalszej części szkolenia uczestnicy zapoznają się z zagadnieniami zaawansowanych sieci, routing, równoważenia obciążenia, QoS, etc. W ostatniej części szkolenia realizowane są tematy związane z modułami raportującymi Eventia. W odróżnieniu od wersji szkolenia R65 wprowadzono tutaj ciekawe ćwiczenia dotyczące raportowania.

Szczegóły można uzyskać pod adresem:

<http://szkolenia.clico.pl/news/szkolenia/szkolenia-autoryzowane/check-point>

#### **Centrum Szkoleniowym Clico:**

email: atc@clico.pl

www: szkolenia.clico.pl

tel. 12-3783700 w.5

#### **Adres centrali:**

Clico Sp.z o.o.

Ul. Oleandry 2

30-063 Kraków

# Budowanie aplikacji internetowych w Aptana Studio

Tworzenie serwisów bogato wykorzystujących technologię Ajax może wymagać umiejętności połączenia różnych języków programowania, bibliotek i frameworków. Sprawne zarządzanie tym melanżem ułatwia nam środowisko Aptana Studio.

Aptana sprawdzi się szczególnie dobrze, jeśli HTML, CSS, DOM, JavaScript, PHP i/lub Ruby on Rails to dla nas codzienność. Środowisko to oferuje również atrakcyjny dodatek – możliwość integracji z Eclipse. Oczywiście Aptana Studio działa również jako w pełni samodzielna aplikacja. Zaletę stanowi wieloplatformowość. Napisana w języku Java Aptana zadziała w każdym spośród popularnych systemów operacyjnych.

## Co potrafi Aptana?

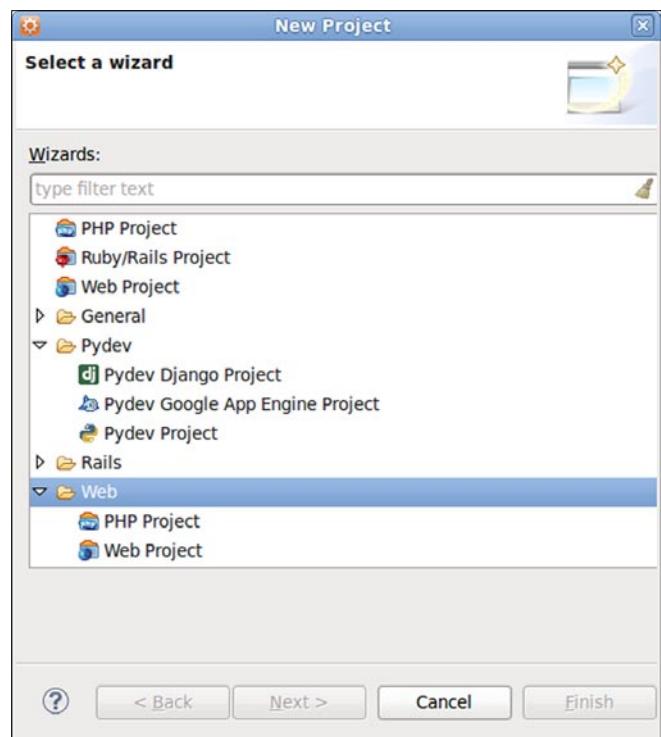
Lista funkcji środowiska jest bardzo rozbudowana. Po pierwsze, efektywnie wspomaga ono tworzenie dokumentów HTML. Podświetlanie składni, uzupełnianie tagów, informacje o obsłudze znaczników przez przeglądarki, formatowanie z użyciem CSS – to tylko najbardziej podstawowe z możliwości, które oferuje nam Aptana. Dalsze ułatwienie stanowią funkcje związane z obiektami DOM. Użytkownik przegląda ich listę uporządkowaną hierarchicznie, filtrując ją i przeskakując błyskawicznie do odpowiednich pozycji w dokumencie.

Opisywany projekt powinien również zainteresować programistów JavaScript. Aptana Studio oferuje wsparcie dla wielu dodatkowych bibliotek i frameworków. Dzięki temu tworzenie aplikacji korzystających z jQuery, Dojo, Ext JS, Mootools, Prototype oraz innych narzędzi JavaScript staje się o wiele łatwiejsze. Aptana sprawdza się także podczas debugowania kodu. Program integruje się z wtyczką Firebug (<http://getfirebug.com/>) przeznaczoną dla przeglądarki Firefox. W efekcie możliwe jest sprawne przetestowanie skryptów i obejrzenie ostrzeżeń oraz komunikatów o błędach w konsoli Firebuga.

Aptana Studio bardzo ułatwia też testowanie tworzonych dokumentów HTML i skryptów w najpopularniej-

szych przeglądarkach. Otwarcie podglądu wymaga tylko jednego kliknięcia. Obsługa FTP pozwala z kolei na szybkie umieszczenie kodu na serwerze.

Zalety środowiska Aptana mogą również docenić programiści PHP, Ruby on Rails, a także Django. Dostępne jest narzędzie RadRails (<http://aptana.org/products/radrails>) – otwarte środowisko wspomagające rozwijanie aplikacji Ruby on Rails. Jeżeli to nie wystarcza, to producent oferuje dodatkową wtyczkę Adobe AIR Plug-In (<http://aptana.org/products/air>), ułatwiającą tworzenie oprogramowania internetowego działa-



Rysunek 1. Każdy znajdzie coś dla siebie – Aptana pozwala na tworzenie projektów PHP, JavaScript, Ruby on Rails, Django oraz innych

jącego także lokalnie. Poza tym dostępne są inne rozszerzenia, dzięki którym uzyskujemy wsparcie dla języka Python oraz widgetów WRT (Nokia Web Runtime). Środowisko znajduje zatem zastosowanie także w pracy nad oprogramowaniem dla urządzeń mobilnych. Użytkownicy zainteresowani dalszą rozbudową Aptany powinni dodatkowo zatrzymać się na forum *3rd Party Plugin Integration* (<http://forums.aptana.com/viewforum.php?f=11&sid=aa37d835a0d9ea71e5666176a6a-0a2d3>). Pokazuje ono, na jak wiele sposobów można wykorzystać opisywane środowisko.

Aktualnie trwają testy nowego wydania Aptana Studio, oznaczonego numerem 3. Ma ono przede wszystkim działać o wiele szybciej. Producent reklamuje również nowe i rozszerzone funkcje – w tym wbudowany terminal i debugger Ruby on Rails, integrację z systemem kontroli wersji Git, a także zwiększone możliwości personalizacji i pełne wsparcie dla HTML5. W naszych testach będziemy wykorzystywali przede wszystkim najnowsze wydanie aplikacji, jednak trzeba pamiętać, że znajduje się ono jeszcze w fazie testów.

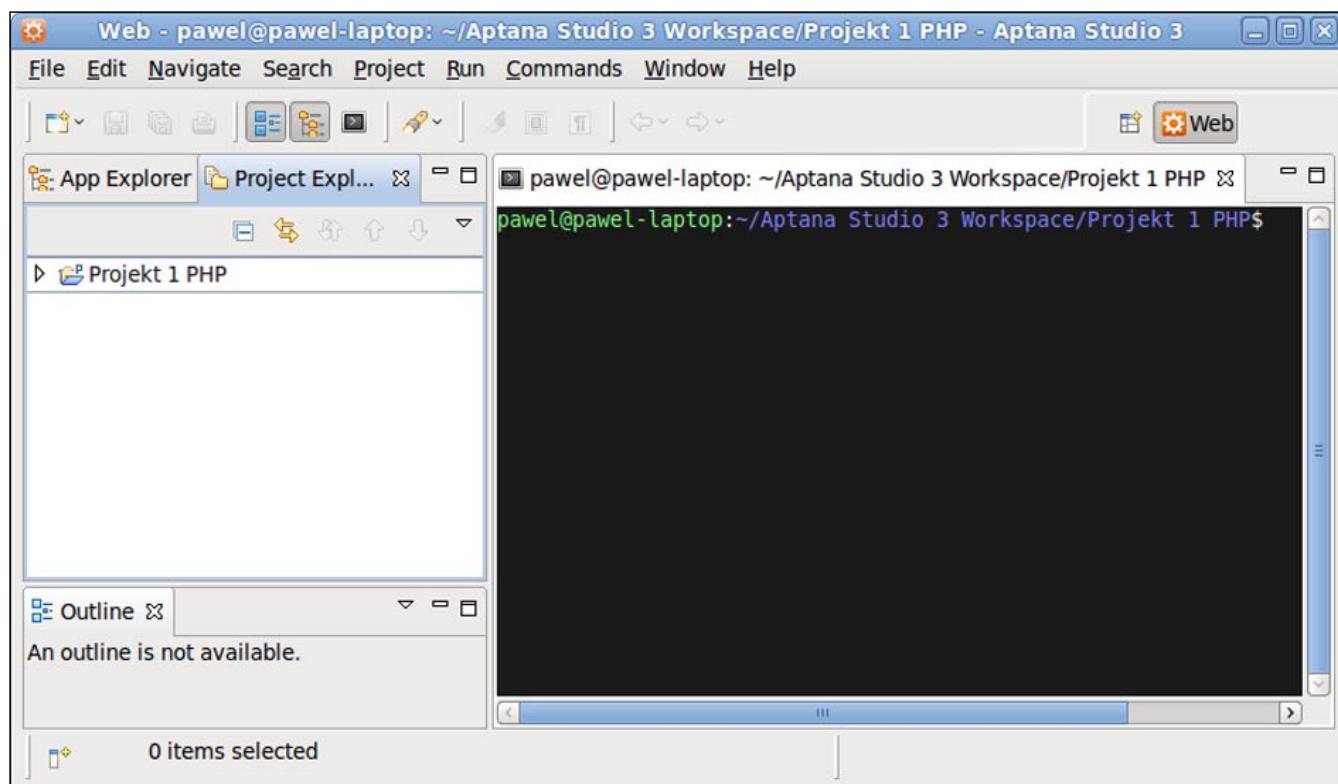
Aptana Studio reklamowane jest jako najlepsze środowisko IDE do tworzenia otwartych aplikacji internetowych. To szumna deklaracja, zwłaszcza jeśli weźmiemy pod uwagę mnogość rozwiązań dostępnych dla programistów PHP/JavaScript/Ruby on Rails. Na korzyść Aptany przemawia ponad 6 milionów pobrań, co stanowi znakomity dowód popularności tego framework'a. Warto więc zapoznać się z jego podstawowymi funkcjami, gdyż potencjalnie może on stać się nieocenioną pomocą w naszej programistycznej pracy. Przy okazji dowiemy się, czy niezbyt skromne deklaracje twórców środowiska nie są przesadzone.

**Instalacja**

Pakiet Aptana współpracuje z aktualnymi wydaniami popularnych dystrybucji Linuksa. Zalecane są Ubuntu oraz Fedora, jednak aplikacja nie nakłada na nas żadnych szczególnych ograniczeń co do wyboru systemu. Ważne jedynie, by był on aktualny – dotyczy to zwłaszcza Aptana Studio 3. Wymagana jest obecność pakietu Java Development Kit w wersji 1.6.x oraz systemu kontroli wersji Git (<http://git-scm.com/>). Potrzebne oprogramowanie można zainstalować z poziomu menedżera paczek wykorzystywanej przez nas dystrybucji.

Oprócz wymienionych, obowiązkowych zależności Aptana korzysta jeszcze z innych, dodatkowych narzędzi. Jeśli zamierzamy zintegrować pakiet ze środowiskiem Eclipse, to konieczne będzie oczywiście jego wcześniejsze zainstalowanie. Użytkownicy chcący rozwijać aplikacje internetowe z użyciem Ruby on Rails muszą natomiast zaopatrzeć się w interpreter języka Ruby, a także pakiety Rails oraz gem.

Aptana Studio pobieramy ze strony domowej projektu w postaci pojedynczego archiwum ZIP. Należy je rozpakować, a następnie uruchomić aplikację polecienniem `./studio3` wydanym w nowo utworzonym katalogu Aptana Studio 3. Żadne dodatkowe procedury instalacyjne nie są potrzebne. Okno programu pojawia się na ekranie monitora po kilku chwilach, w zależności



Rysunek 2. Aptana Studio 3 tuż po uruchomieniu i utworzeniu nowego projektu

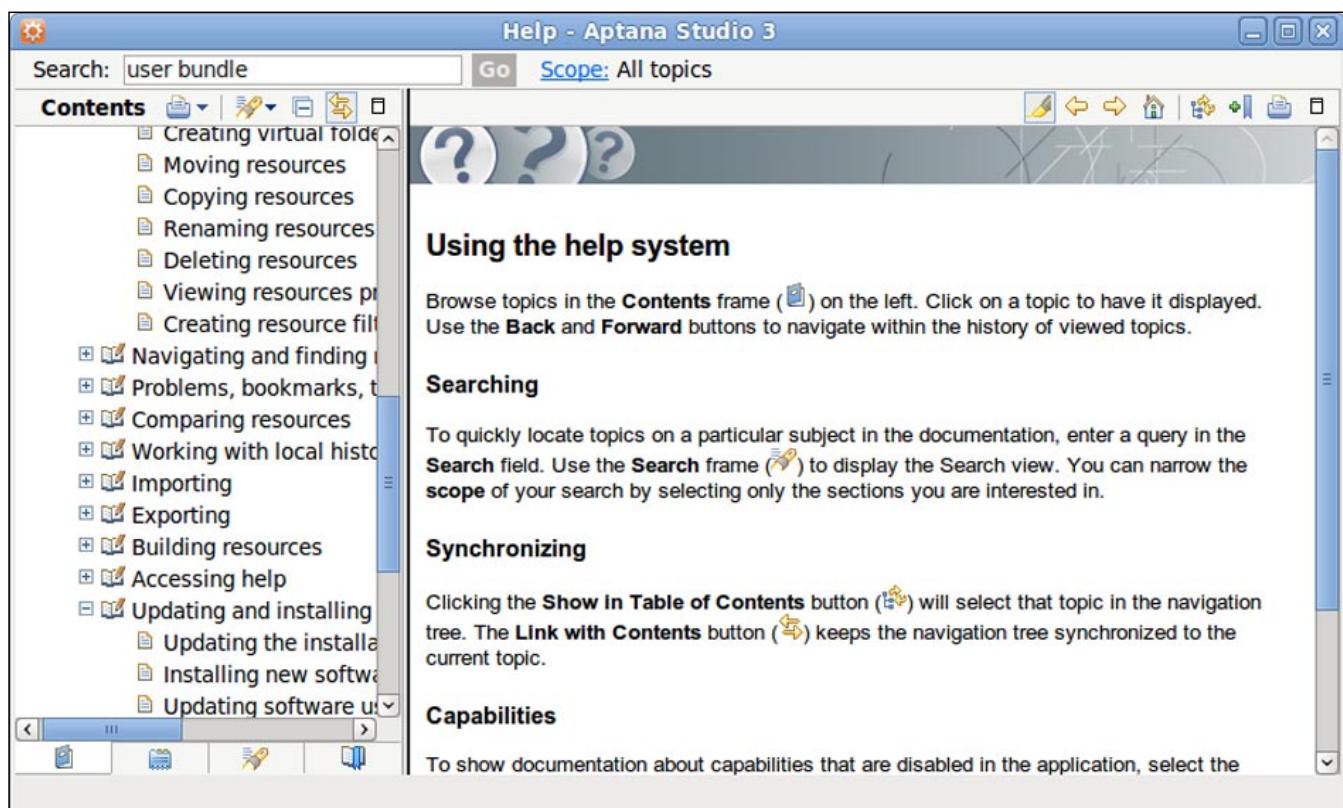
od mocy obliczeniowej komputera. Zgodnie z zapowiedziami twórców środowiska nowe wydanie Aptany powinno być zdecydowanie szybsze. Nie oczekujmy jednak znaczającej poprawy wydajności. Aplikacja działa w miarę szybko, o ile pominiemy trochę zbyt długi czas ładowania. W podobny sposób zachowywało się jednak także Aptana Studio 2.0. Użytkownicy, którzy wzięli sobie do serca zapowiedzi producenta środowiska mogą poczuć się zawiedzeni. Oczekiwania związane z poprawą wydajności nie powinny być zatem podstawową przesłanką motywującą nas do aktualizacji oprogramowania.

W nieco inny sposób instalujemy wtyczkę dla Eclipse. Nie powinniśmy pobierać żadnego pakietu z witryny Aptana Studio, a jedynie przejść na stronę pozwalającą na ściagnięcie wybranej przez nas wersji środowiska (2.0.x lub 3 Beta). Uaktywnijmy tam opcję *Eclipse Plug-in Version*. Przełącznik *Standalone Version* zostanie wówczas automatycznie odznaczony. Po naciśnięciu przycisku rozpoczęjącego pobieranie zostaniemy przekierowani do strony, gdzie znajduje się link do wtyczki. Należy go skopiować i wkleić w okienku *Install New Software* w środowisku Eclipse. Próba pobrania pakietu zawierającego wtyczkę bezpośrednio z poziomu okna przeglądarki spowoduje wyświetlenie ostrzegawczego komunikatu. Jeżeli decydujemy się na wykorzystanie Aptany razem z Eclipse, to instalację wtyczki powinniśmy wykonywać wyłącznie z poziomu drugiego z wymienionych środowisk.

## Początek pracy

Po uruchomieniu aplikacji i krótkim oczekiwaniu na ekranie pojawi się główne okno programu. Jest ono podzielone na co najmniej dwa oddzielne panele. W lewej kolumnie znajdują się dwie zakładki: *App Explorer* oraz *Project Explorer*. Pierwsza z nich wyświetla zawartość konkretnego projektu. Z kolei karta *Project Explorer* umożliwia przejrzenie listy wszystkich rozwiązywanych aplikacji, usuwanie ich oraz tworzenie nowych. Po pierwszym uruchomieniu Aptana Studio powinniśmy oczywiście rozpocząć pracę od założenia pustego projektu. Możemy tego dokonać zarówno z poziomu zakładki *Project Explorer*, jak i głównego menu. W drugim przypadku konieczne jest wydanie polecenia *File > New > Project*. Otworzy ono osobne okienko, w którym powinniśmy wybrać jeden spośród wielu typów wspieranych projektów.

Aptana Studio jest środowiskiem uniwersalnym, pozwalającym na rozwijanie oprogramowania i serwisów wykorzystujących kilka z najpopularniejszych technologii internetowych. Otwarte przez nas przed chwilą okienko tworzenia nowego projektu umożliwia rozpoczęcie pracy między innymi nad aplikacjami opartymi na PHP, Ruby on Rails, Django oraz Google App Engine. Niektóre z projektów można utworzyć bezpośrednio z poziomu głównego menu Aptany, bez konieczności przechodzenia do osobnego okna. Potrzebne polecenia (w tym *PHP Project* oraz *Ruby/Rails Project*) znajdują się w podmenu *File > New*.



Rysunek 3. Studio 3 posiada rozbudowany system pomocy

Wybór jednego z typów projektu otwiera okno kreatora. Wprowadzamy w nim nazwę aplikacji, jej lokalizację i ewentualnie dodatkowe informacje. Przykładowo decydując się na utworzenie projektu PHP należy określić wersję interpretera, z którą będzie on kompatybilny. Domyślna wartość to PHP 5.3.x.

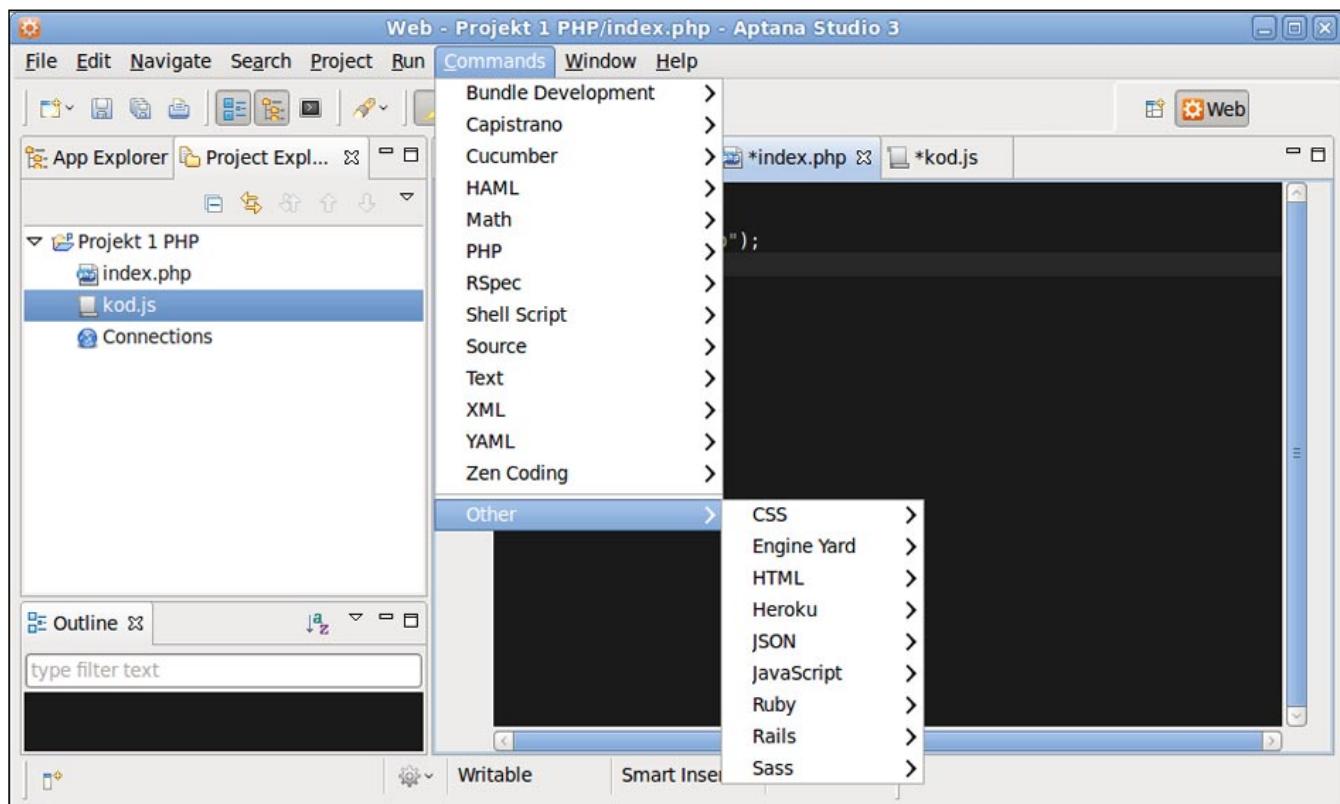
Zestaw funkcji związanych z zarządzaniem całymi projektami jest o wiele szerszy. Kliknięcie lewym przyciskiem myszy na nazwie projektu (wyświetlanej na zakładce *Project Explorer*) otwiera obszerne kontekstowe menu. Znajdziemy w nim między innymi komendy służące do tworzenia pojedynczych plików wchodzących w skład aplikacji (*New > File*). Poza tym to samo kontekstowe menu umożliwia przesyłanie danych na serwer internetowy (*File Transfer > Upload* oraz *File Transfer > Download*). Kolejny zestaw funkcji pozwala na otwarcie plików w edytorech kodu źródłowego. Potrzebne komendy znajdują się w podmenu *Open With*. Dopuszczalne jest między innymi otwarcie wbudowanego edytora kodu PHP/JavaScript (*Aptana PHP Source Editor*, *JavaScript Source Editor*), edytora tekstowego (*Text Editor*), a także zewnętrznych narzędzi (*System Editor*). Wszystkie dokumenty można kopiować, wklejać, eksportować, a także importować do projektu z zewnątrz.

Edytor kodu źródłowego obecny w Aptanie posiada wszystkie podstawowe funkcje, w które powinno być wyposażone każde przyzwoite narzędzie tego typu. Mamy więc wsparcie dla różnych kodowań znaków, numerację linii, podświetlanie składni, informa-

cje o błędach. Bardzo rozbudowany jest system podpowiedzi, który znaczco ułatwia pisanie kodu HTML, CSS, PHP oraz Ruby on Rails. Niestety, w najnowszej wersji aplikacji funkcja ta nie działa w pełni bezproblemowo. Niektórzy z użytkowników sygnalizują brak automatycznych uzupełnień, zwłaszcza podczas tworzenia kodu JavaScript. Problemy takie nie pojawiają się we wcześniejszych wydaniach Aptany. Chociaż część trudności związanych z systemem podpowiedzi można przezwyciężyć, to jednak stanowią one przesłankę przemawiającą za odłożeniem migracji do najbliższego wydania środowiska aż do momentu pojawienia się stabilnej wersji.

Oprócz edytora kodu źródłowego, pojawiającego się po prawej stronie okna, Aptana Studio 3 udostępnia również terminal. Wyświetlany jest on razem z edytorem, jako osobna zakładka.

Aptana Studio obsługuje spory zestaw technologii internetowych. Wsparcie nie ogranicza się do podświetlania składni i autouzupełniania. Możliwe jest wprowadzanie do plików zarówno szablonów pętli, jak i podstawowych funkcji. Dotyczy to nie tylko samych języków programowania, ale także dodatkowych bibliotek oraz frameworków. Pełną listę możliwości ujrzymy po kliknięciu prawym klawiszem myszy wewnątrz edytora kodu. Otworzy to rozbudowane menu kontekstowe. Biblioteki znajdują się w podmenu *Commands*. Producent środowiska zadbał o wsparcie dla najważniejszych technologii – tagów HTML, stylów CSS, funkcji PHP, JavaScript i Ruby. Są też jednak inne, bardziej



Rysunek 4. Menu Commands środowiska Aptana

wyspecjalizowane biblioteki, obsługujące między innymi język XML oraz standard JSON. Programistów Ruby on Rails ucieśnie zapewne wsparcie dla narzędzi Capistrano (<http://github.com/capistrano/capistrano/wiki/Documentation-v2.x>) i Cucumber (<http://cukes.info/>).

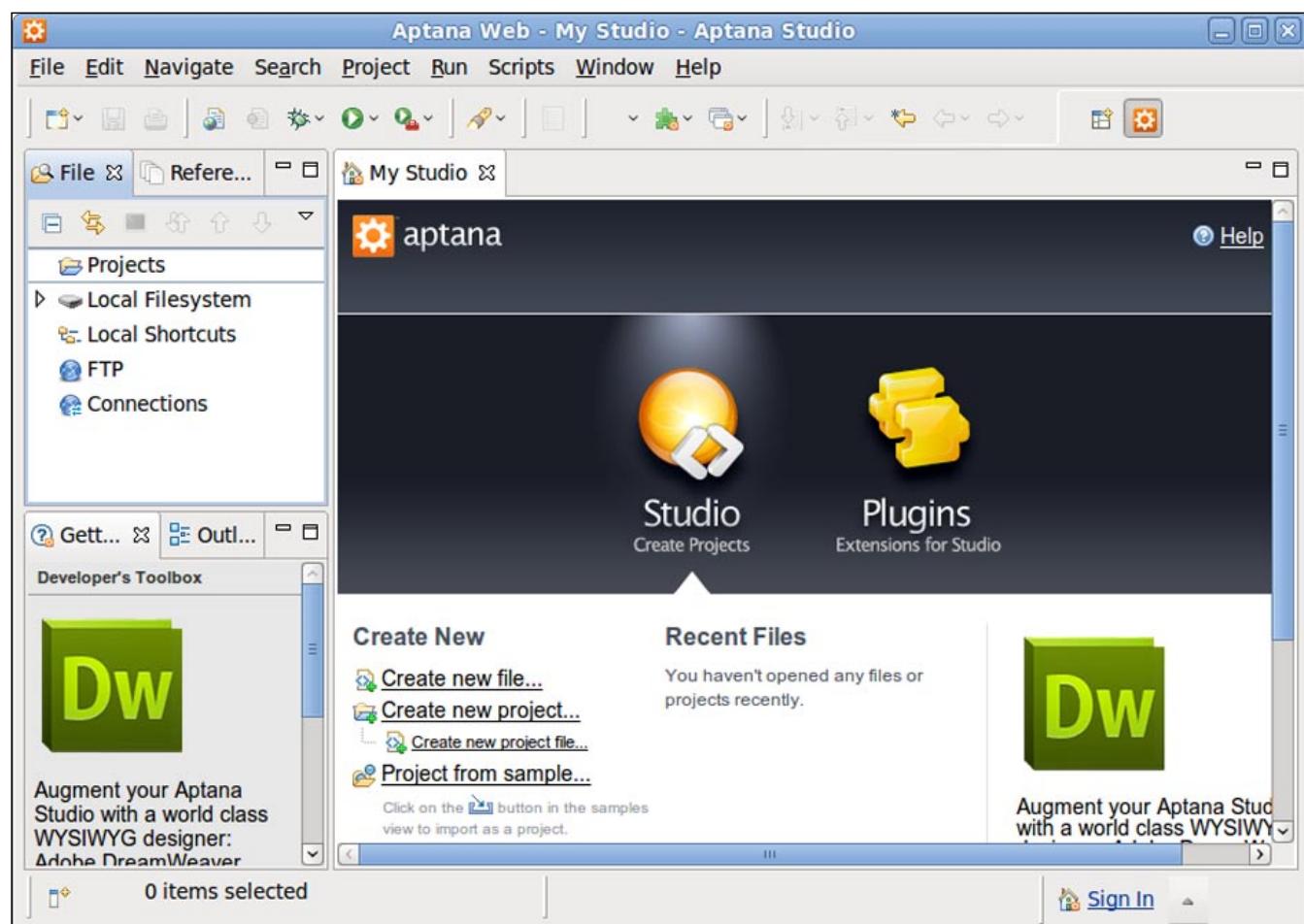
Bibliotekę komend, wyświetlającą w menu *Commands*, można rozbudowywać. Nowe pakiety należy instalować polecienniem *Commands > Bundle Development > Install Bundle*, które znajdziemy we wskazanym wcześniej menu kontekstowym. Komenda ta otwiera okienko *Select Bundle To Install*. Znajdziemy tam między innymi pakiety jQuery, Mercurial, Wordpress, GitHub oraz kilka innych.

## Zarządzanie projektem

Zarządzanie stale powiększającym się projektem ułatwia zestaw funkcji służących do importu i eksportu danych, interakcji z systemem kontroli wersji Git oraz przesyłania plików na serwery FTP. Kompletne projekty można przenosić do Aptany za pomocą komendy *Import* z menu *File*. Dane źródłowe pochodzą z systemu plików, ze spakowanych archiwów, a także repozytoriów Git. Eksport danych też jest możliwy (poleciennie *File > Export*). Dokumenty trafiają do systemu plików lub do archiwów – w zależności od potrzeb użytkownika.

Transferem całych projektów oraz poszczególnych fragmentów kodu źródłowego zarządzamy za pomocą funkcji połączeń (*Connections*). Aptana Studio pozwala na zdefiniowanie ustawień decydujących o tym, gdzie trafią lub skąd zostaną pobrane dane źródłowe. Dzięki temu kolejne aktualizacje będzie można natychmiast wprowadzić na serwer internetowy, do repozytorium, spakować w archiwum, lub umieścić na nośniku pełniącym rolę kopii zapasowej. Połączenia definiujemy z poziomu zakładki *Project Explorer*, za pomocą komendy *File Transfer > Connections*. Poleciennie to otwiera okienko *Connection Manager*. Po utworzeniu (ikoną plusa) nowego połączenia wprowadzamy jego nazwę (*Name*), źródło danych (ramka *Source*) oraz lokalizację docelową (*Destination*). Pliki można przesuwać pomiędzy projektami, lokalizacjami oraz serwarami. Zestaw wszystkich potrzebnych połączeń warto przygotować od razu, by podczas kolejnych sesji z Aptana Studio importować i eksportować dane pojedynczym kliknięciem myszy, bez konieczności odszukiwania katalogów na dysku i wpisywania adresów serwerów FTP.

Połączenia nie są jedyną rzeczą, którą warto przygotować sobie już na początku pracy z projektem. Zadbajmy również o poprawne skonfigurowanie interpre-



Rysunek 5. Główne okno Aptany w wersji Studio 2

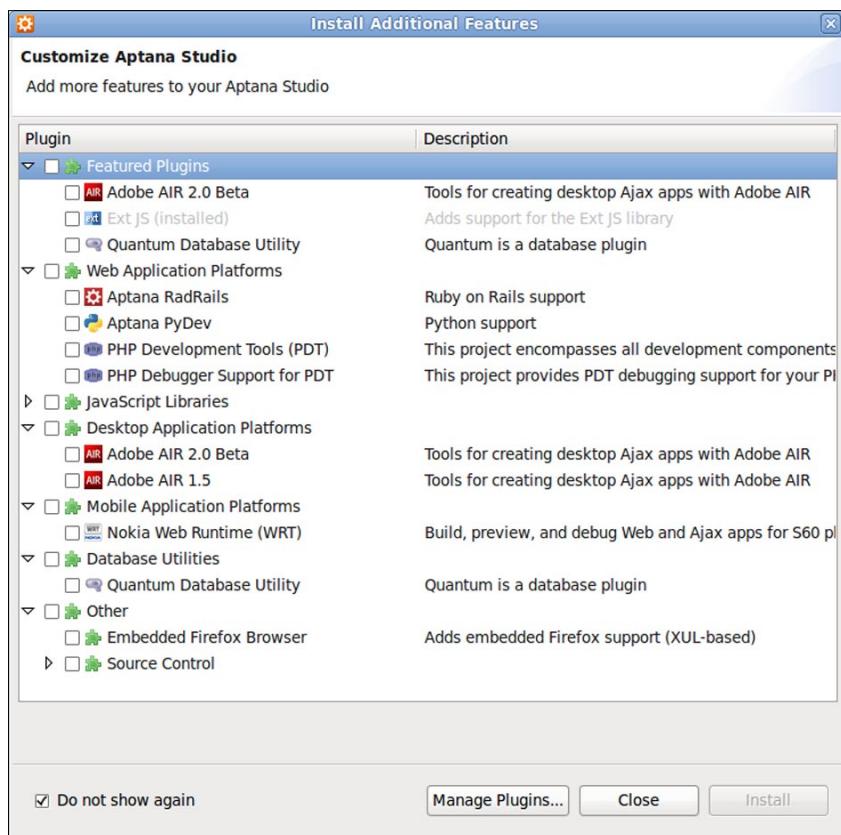
terów wykorzystywanych podczas uruchamiania aplikacji. Dotyczy to przede wszystkim tych użytkowników, którzy zamierzają korzystać z Django. Powinni oni wydać polecenie *Run As > Run Configurations*, a następnie zaznaczyć potrzebną konfigurację, kliknąć prawym klawiszem myszy i wybrać *New*. Pozwoli to na wprowadzenie własnych ustawień.

Aptana Studio 3 działa bardzo stabilnie. Niekiedy trudno zauważać, że mamy do czynienia z oprogramowaniem znajdującym się wciąż w fazie beta. Nie oznacza to jednak, że trudności w ogóle się nie zdarzają. Niektórzy użytkownicy wersji testowej wskazywali na problemy z uruchomieniem środowiska – i to we wspieranej oficjalnie dystrybucji Ubuntu. Podczas testów nie pojawiły się jednak żadne trudności tego typu.

Nieco do życzenia pozostawiają kwestie związane z użytecznością oraz intuicyjnością oprogramowania. Niekiedy trudno jest odnaleźć funkcje, który były obecne w Aptana Studio 2, natomiast w najnowszym wydaniu zostały „ukryte”. Niewykluczone, że część z tych kłopotów wynika z przyzwyczajeń. Nie da się bowiem ukryć, że jeśli mamy za sobą dużą pracę z wcześniejszymi wersjami Aptany, to odnalezienie się w zaktualizowanej aplikacji może nam sprawić problemy, na szczęście niewielkie. Nie sposób też pozbyć się wrażenia, że najnowsze wydanie środowiska jest mniej przyjazne niż poprzednie. Zmiany, które wprowadzono w interfejsie, nie są jednak drastyczne, stąd też po czasie poświęconym na zapoznanie się z aplikacją można powrócić do normalnej, równie wydajnej pracy. Na korzyść Aptany przemawia też fakt poważnego traktowania przez producenta uwag użytkowników. Przynajmniej niektóre z wykrytych błędów powinny zostać usunięte w finalnej wersji opisywanego środowiska.

## Aptana Studio 2

Jeśli mimo wszystko nie zdecydujemy się na skorzystanie ze Studia 3, albo zamierzamy odłożyć tę decyzję na później, do momentu pojawienia się stabilnego wydania, to wciąż możemy skorzystać z Aptana Studio 2. Warto pokusić się o wypróbowanie tego środowiska, gdyż jest ono bardzo proste w obsłudze, a zestaw funkcji jest niemal równie bogaty, jak w najnowszej, przetestowanej przed chwilą wersji. Instalacja nie sprawia żadnych trudności i przebiega w identyczny sposób, jak w przypadku Studia 3. Potrzebne pakiety znajdziemy na stronie produktu. Dostępna jest także wtyczka dla



Rysunek 6. Instalacja dodatków w Aptana Studio 2

środowiska Eclipse. Wymagania programowe nie odiegają od opisanych uprzednio dla Studio 3.

Po pierwszym uruchomieniu Aptana Studio 2 pojawia się okno podzielone na dwa panele. W lewym znajdują się menedżer plików oraz projektów. W nowej wersji aplikacji zmieniło się zatem niewiele. Nieco inaczej wygląda natomiast prawy panel. Początkowo zawiera on kilka przycisków ułatwiających rozpoczęcie pracy z programem. Poza tym panel ten – w którym docelowo użytkownik edytuje kod tworzonego projektu – ułatwia dostęp do wtyczek rozszerzających funkcjonalność całego środowiska. W dodatkach drzemie ogromna siła. W Aptana Studio 2 są one bardzo łatwo dostępne. Wystarczy skorzystać z narzędzia *Plugins Manager*, które pozwala na doinstalowanie wielu dodatkowych bibliotek.

Lista rozszerzeń Studio 2 może przyprawić o zawrót głowy. Właśnie w oknie menedżera wtyczek znajdziemy wspomniany wcześniej pakiet RadRails (wsparcie dla Ruby on Rails), a także Aptana PyDev (obsługa Pythona). Poza tym otrzymujemy również narzędzia do tworzenia aplikacji Adobe AIR i oprogramowania mobilnego (Nokia Web Runtime). Zadowoleni będą też programiści JavaScript. Aptana Studio 2 pozwala na natychmiastowe pobranie i zainstalowanie niemal wszystkich z najpopularniejszych bibliotek. Na rozwijanej liście *JavaScript Libraries* znajdziemy między innymi Dojo, ExtJS, jQuery, MochiKit, MooTools, Prototype, Scriptaculous oraz kilku innych.

Aby otworzyć okno menedżera wtyczek należy wydać polecenie *Window > Show View > Other*. Otworzy ono okienko *Show View*. Następnie powinniśmy wybrać (z rozwijanej listy) *Aptana Views > Plugins Manager*. Kliknięcie przycisku z ikoną plusa (*Install new feature*) otworzy listę wtyczek. Trzeba wówczas zaznaczyć interesującą nas bibliotekę i nacisnąć przycisk *Install*. Pojawi się okienko z prośbą o zaznaczenie ewentualnych dodatkowych pakietów (przykładowo innych bibliotek JavaScript). Wszystkie dokonane zmiany zwierdzamy za pomocą przycisku *Next*. Rozpoczyna on proces określania zależności. Zostanie on zakończony prośbą o potwierdzenie, a potem – o zaakceptowanie licencji. Następnie aplikacja przystąpi do instalacji dodatków. Po zakończeniu środowisko zostanie zrestartowane.

## Podsumowanie

Funkcje opisane w tym artykule nie wyczerpują oczywiście wszystkich możliwości Aptana Studio. Oprócz

wspomnianych ułatwień na etapie tworzenia aplikacji internetowych (uzupełnianie kodu, łatwy dostęp do bibliotek i frameworków) środowisko jest bardzo pomocne także w trakcie testowania i debugowania (wsparcie dla wtyczki Firebug, podgląd w najpopularniejszych przeglądarkach). Najnowsze wydanie nie stanowi rewolucji, jednak tylko dlatego, że już w Studio 2 zestaw funkcji był naprawdę imponujący.

Niezależnie od tego, na którą wersję Aptana Studio zdecydujemy się, że pewność nie będziemy żałować. Opisywane środowisko sprawdza się świetnie zwłaszcza podczas pracy nad projektami opartymi na JavaScript oraz Ruby on Rails. Sporo zyskają także użytkownicy tworzący kod PHP, choć w ich przypadku Aptana może okazać się narzędziem zbyt zaawansowanym w stosunku do potrzeb. Wymagania sprzętowe i programowe nie są wygórowane, aplikacja nie powinna też sprawiać problemów w trakcie instalacji i użytkowania. Środowisko Aptana Studio ma zatem szansę stać się dla nas narzędziem ułatwiającym tworzenie efektownych aplikacji internetowych.

## W Sieci

- Aptana Studio 2 – <http://aptana.org/products/studio2>;
- Aptana Studio 3 – <http://aptana.org/products/studio3>.

REKLAMA

## Wybierz swój nowy system, poznaj Linuksa krok po kroku!

### wortalu:

- recenzje dystrybucji**
- porównanie funkcji**
- opis instalacji**
- i dużo więcej**

Twoje pierwsze źródło  
wiedzy o Linuksie

ponad ćwierć miliona  
odwiedzin miesięcznie!



**jakilinux.org**

kontakt / reklama  
[borys.musielak@gmail.com](mailto:borys.musielak@gmail.com)

Pierwsza wirtualna konferencja w Polsce

# OPEN SOURCE DAY ONLINE

Serdecznie zapraszamy do udziału w **pierwszej w Polsce całkowicie wirtualnej** konferencji poświęconej rozwiązaniom Red Hat/JBoss i Open Source, która odbędzie się już

**18 LISTOPADA 2010**

## **RED HAT ENTERPRISE LINUX 6**

wiadomości z pierwszej ręki!

Udział Gości Specjalnych z firmy **Red Hat**, m.in. **Frank Weyns**

(Principal Solution Architect) i **Phil Andrews** (Regional Director EMEA).

Zapraszamy do wysłuchania wykładów światowych ekspertów na temat budowy i infrastruktury Cloud, wirtualizacji, nowej wersji Satellite Server oraz bezpieczeństwa w wirtualnym świecie.

Mogą Państwo do nas dołączyć z dowolnego miejsca - **wystarczy rejestracja i dostęp do internetu**. W programie przewidziany **czat na żywo** z prelegentami oraz ciekawe konkursy z nagrodami.

Agenda spotkania dostępna na stronie konferencji: [www.OpenSourceDay.pl](http://www.OpenSourceDay.pl)

**Udział w konferencji jest bezpłatny.**

Zapraszamy do rejestracji!



**WWW.OPENSOURCEDAY.PL**

# Statyczne systemy CMS

Jedną z niezaprzeczalnie mniej licznych grup systemów zarządzania treścią jest grupa systemów generujących statyczną treść. Dzięki temu, że pliki generowane są statycznie, możemy używać tego typu CMS-ów na naszej domowej maszynie, a dopiero później synchronizować zmiany. Takie podejście pozwala wyeliminować wiele wad dynamicznych systemów treści.

## I. Wstęp

W dzisiejszych czasach wykorzystywanie systemów CMS (ang. *Content Management System*) stało się na tyle powszechnie, że trudno wyobrazić sobie powrót do czasów, gdy strony internetowe były statyczne, a ich aktualizacja oraz rozbudowa potrafiła być czasochłonna, problematyczna oraz wymagała znajomości co najmniej HTML-a. Pojawienie się systemów zarządzania treścią było przejściem do nowej epoki. Epoki, która umożliwiła głównie osobom „nietechnicznym” łatwą aktualizację oraz rozbudowę stron WWW, zostawiając prezentację danych oprogramowaniu.

### Argumenty przeciw dynamicznym CMS-om

Niestety wśród wielu niezaprzeczalnych zalet systemów CMS znajduje się kilka łyżek dziegciu. Do jednej z największych wad należą duże wymagania. Oprogramowanie takie jak MySQL i PHP to nierzadko zbyt wiele dla serwerów uczelnianych oferujących studentom często tylko czysty serwer WWW.

Ponadto wydajność dynamicznych generatorów treści wraz z lawinowo rosnącą liczbą użytkowników potrafi spaść do punktu, w którym cały system przestaje albo działać, albo nasz hoster wypowiada nam umowę. Częstą techniką, która ma zapobiec tego typu sytujom, jest „keszowanie” (ang. Cache)m czyli w tym

#### Od autora:

Pierwotnie artykuł miał skupić się tylko na NanoBloggerze. Jednak po rozmowie z Pawłem Oniszczukiem postanowiłem mocno go rozszerzyć i podzielić na dwie części.

- Część pierwsza omawia zagadnienie statycznych CMS-ów oraz NanoBloggera.
- Część druga skupi się na Jekyll.

wypadku generowanie statycznych plików HTML przez dynamiczny z założenia CMS.

Mimo wszystko do największych bolączek tego typu oprogramowania należy bezpieczeństwo. Nikt nie powinien mieć złudzeń, że ten czy inny system jest w 100% bezpieczny. Zwłaszcza że dziury w systemach zarządzania treścią odnajdywane są praktycznie codziennie. Ponadto większość z dostępnych na rynku CMS-ów oferuje możliwość instalacji rozszerzeń lub wtyczek, które to dodatkowo przyczyniają się do obniżenia bezpieczeństwa całości.

### Statyczny, ale wciąż CMS

Jedną z niewątpliwie mniej licznych grup systemów zarządzania treścią jest grupa systemów generujących statyczną treść. Dzięki temu, że pliki generowane są statycznie, możemy używać tego typu CMS-ów na naszej domowej maszynie, a dopiero później synchronizować zmiany. Takie podejście pozwala wyeliminować wyżej opisane wady dynamicznych systemów treści, przy minimalnym wykorzystaniu zasobów, gdyż wystarczy nam tylko czysty serwer WWW. Dzięki czemu tego typu CMS-ów możemy używać ze znacznie większym powodzeniem w środowisku, w którym dynamiczne CMS-y sobie po prostu nie radzą.

Niestety ta grupa oprogramowania CMS mimo wielu zalet jest mniej znana, a przez to mniej popularna. Za taki stan rzeczy należy winić kilka stereotypów, które postaram się obalić.

#### Walka ze stereotypami

Jednym z najczęściej zarzucanych niedoskonałości statycznych CMS-ów jest brak pracy kilku użytkowników jednocześnie. Na szczęście dzięki systemom kontroli wersji takim jak git<sup>1</sup>, nie powinniśmy mieć żadnych pro-

<sup>1</sup>. Więcej informacji o systemie kontroli wersji git - [https://secure.wikimedia.org/wikipedia/pl/wiki/Git\\_\(oprogramowanie\)](https://secure.wikimedia.org/wikipedia/pl/wiki/Git_(oprogramowanie))

blemów z rozproszoną pracą nad naszą stroną. W tym miejscu należy wspomnieć o Jekyll<sup>2</sup>, który jest generatorem statycznej treści wykorzystanym przez twórców Github-a<sup>3</sup> w roli domyślnego oprogramowania zarządzającego stronami hostowanymi tam projektów.

Kolejną z półprawd, która częściowo pokrywa się z faktycznym stanem rzeczy, jest brak możliwości dodawania przez użytkowników treści takiej jak komentarze. Jak wspomniałem, taki pogląd częściowo pokrywa się z prawdą, gdyż bez problemu możemy zintegrować naszą stronę z takimi rozwiązaniami jak IntenseDebate<sup>4</sup> i przechowywać komentarze w chmurze obliczeniowej. Warto wspomnieć, że z tego typu rozwiązań coraz częściej korzysta się również w dynamicznych systemach CMS, by choć trochę odciążyć serwery.

Dobra wyszukiwarka to często jedyny klucz do odnalezienia interesującej nas treści. Wielu powie, że zaopatrzenie statycznej przecież strony w wyszukiwarkę jest niemożliwe. Nie należy zapominać jednak, że dzięki oferowanym przez firmę Google narzędziom możemy bez najmniejszego problemu zaopatrzyć nasz statyczny CMS w pełnoprawną wyszukiwarkę najwyższych lotów.

Pamiętajmy również, że jakiekolwiek błędy oraz wady statycznego systemu zarządzania treścią nie będą miały wpływu na bezpieczeństwo danych.

## II. NanoBlogger

Gdyby ktoś wcześniej powiedział mi, że używa CMS-a napisanego w powłoce systemowej, nie potraktowałbym tego poważnie. Jak się jednak okazuje, skrypty powłoki to nie tylko pomocne i ułatwiające pracę głównie administratorom małe programy, ale nieradko bardziej złożone rozwiązania czy nawet gry<sup>5</sup>.

Po takim wstępnie nie powinno już nikogo dziwić dalsza część artykułu poświęcona napisanemu w Bashu – NanoBloggerowi, którego sam z powodzeniem używam do prowadzenia uczelnianej strony.

### How it's Made?

Ten sprytny skrypt oprócz wspomnianego Bash-a wykorzystuje między innymi takie narzędzia powłoki jak cat, grep, sed czy wszechpotężny vi - choć tego ostatniego możemy ostatecznie zmienić na inny ulubiony konsolowy edytor.

Warto w tym miejscu również zauważać, iż cała baza danych domyślnie oparta jest na plikach txt, z których dopiero później generowane są odpowiednie statyczne pliki html oraz xml.

Czytelnik powinien być świadomy, że skrypty powłoki mają to do siebie, iż nie cechuje ich wysoka wydajność.

Dlatego NanoBlogger nie jest dobrym rozwiązaniem dla osób, które chcieliby wykorzystać go do prowadzenia większych projektów.

### Co to w ogóle potrafi?

NanoBlogger, jak na system zarządzania treścią przystało, umożliwia wszystko to, co każdy inny szanujący się CMS. Bez problemu możemy więc dodawać wpisy czy posegregować je w odpowiednich kategoriach. Możemy utrzymywać archiwum czy korzystać z plug-inów. NanoBlogger bez problemu obsługuje RSS-y w wersji pierwszej oraz drugiej, a także Atom-a. Wpisy możemy dodawać w HTML-u lub po zainstalowaniu odpowiedniej wtyczki w Markdown<sup>6</sup>.

### Jak zacząć?

Tym, co zasługuje na duży plus, jest brak konieczności instalowania czegokolwiek. Wystarczy, że pobierzemy najnowszą wersję NanoBloggera i wypakujemy ją na przykład do katalogu bin w katalogu home, dzięki czemu możemy z łatwością wykorzystać NanoBloggera na serwerze WWW, do którego mamy dostęp przez SSH – warunkiem koniecznym będzie oczywiście możliwość uruchamiania na serwerze własnego oprogramowania. W dalszej części artykułu poświęconej konfiguracji dowiemy się, w jaki sposób skonfigurować tego CMS-a, by mimo instalacji lokalnej automatycznie aktualizował odpowiednie pliki na serwerze (patrz Dodatek A), dzięki czemu nie będziemy musieli za każdym razem przenosić plików ręcznie.

Wspomniałem wcześniej, że wystarczy wypakować NanoBloggera do katalogu domowego, co jest bardzo

#### **Listing 1. Instalacja w katalogu domowym**

```
#Pobieramy NanoBloggera
wget http://nanoblogger.sourceforge.net/downloads/
nanoblogger-3.4.2.tar.gz

#Wypakowujemy
tar -xf nanoblogger.3.4.tar.gz
##Dowiązanie do ~/bin
ln -s nanoblogger.3.4 ~/bin/nb
#Dopisujemy do zmiennej $PATH ścieżkę do naszej
instalacji NB:
echo "PATH=$PATH:~/bin/nb" >> ~/.bashrc
```

#### **Listing 2. Pierwsze uruchomienie**

```
nb -b /<ścieżka>/<do>/<bloga>/ add weblog
```

2. Jekyll zostanie poświęcona druga część artykułu.

3. <http://github.com>

4. <http://intensedebate.com>

5. Tetris w sed <http://thecamels.org/2010/01/15/tetris-w-sedzie/>

6. <https://secure.wikimedia.org/wikipedia/pl/wiki/Markdown>

przydatne, gdy poziom uprawnień w systemie uniemożliwia nam instalację oprogramowania. Warto w tym miejscu od razu dodać do zmiennej \$PATH ścieżkę do katalogu /home/<user>/bin/nb, dzięki czemu będziemy mogli wywołać NanoBloggera przez użycie komendy nb bez konieczności podawania pełnej ścieżki dostępu. Wszystkie kroki związane z procesem instalacji zostały opisane w Listingu 1.

## Pierwszy start i konfiguracja

Teraz jesteśmy już gotowi do pierwszego uruchomienia oraz konfiguracji. W tym miejscu warto wspomnieć, iż NanoBlogger posiada dwa pliki konfiguracyjne. Pierwszy to -nb.conf; umożliwia on skonfigurowanie wielu globalnych ustawień, dzięki którym możemy sterować sposobem działania NanoBloggera. Plik ten znajdziemy w głównym katalogu programu. Warto wspomnieć o możliwości użycia lokalnej kopii tego pliku - /home/<user>/nb.conf. Zawarta w nb.conf konfiguracja pozwala na przykład zdefiniować domyślny katalog, w którym każdy z blogów będzie domyślnie przechowywany. Za pomocą tego pliku możemy również zmienić język.

Drugi rodzaj plików konfiguracyjnych znajduje się w katalogu każdego z utworzonych blogów. Pliki blog.conf pozwalają skonfigurować indywidualne ustawienia każdego bloga. Przykładowa lista wybranych opcji znajduje się w Listingu 3. Zanim jednak przedziemy do konfiguracji naszego bloga, musimy go najpierw utworzyć. Aby to zrobić, musimy wydać polecenie przedstawione w Listingu 2.

Przełącznik -b informuje NanoBloggera o ścieżce dostępu do naszego bloga. Ważne jest, by podczas tworzenia nowego bloga nie podawać ścieżki do katalogu, który już istnieje. Po wydaniu tej komendy zostaniemy poproszeni o dokonanie zmian w pliku konfiguracyjnym. Każ-

da z opcji została dokładnie opisana wraz z przykładową konfiguracją. W razie jakichkolwiek problemów w każdej chwili możemy zajrzeć do dokumentacji.<sup>7</sup>

Konfigurację możemy w każdej chwili zmienić, edytując ręcznie plik blog.conf lub wydając polecenie z Listingu 4.

Teraz, gdy nasz blog został już utworzony, możemy zająć się dodawaniem treści. Sposób redagowania i katalogowania wpisów oraz kilku innych popularnych czynności zostało omówionych w Listingu 5.

## Poszerzamy horyzonty - style & wtyczki

Jak wspomniałem wcześniej, Nanoblogger jest platformą, którą bez problemu możemy rozbudować dzięki wtyczkom oraz dopasować do swoich upodobań dzięki stylom. Domyślnie Nanoblogger zaopatrzony jest w dwa style oraz kilka najpopularniejszych wtyczek. Jeśli chcielibyśmy rozbudować go o dodatkowe możliwości lub skorzystać z dużo ciekawszych stylów, powinniśmy zainteresować się pakietem nanoblogger-extra<sup>9</sup>.

### Style

Cała strona prezentacyjna naszego bloga budowana jest na podstawie plików htm znajdujących się w katalogu templates oraz kaskadowych arkuszy stylów (ang. Cascading Style Sheets, CSS). Nie ma tu, jak widać, żadnej filozofii. Toteż jeśli domyślny styl oraz style z pakietu extra nam nie pasują, łatwo możemy zmodyfikować lub stworzyć styl odpowiadający naszym upodobaniom<sup>10</sup>. Aby zmienić styl, należy zmodyfikować wpis BLOG\_CSS="" w pliku blog.conf.

### Wtyczki

Oczywiście jest, że opisanie wszystkich wtyczek przeciąga tę publikację. Dlatego też postanowiłem opisać trzy najbardziej przydatne moim zdaniem wtyczki. Jeśli

### Listing 3. Wybrane ustawienia lokalnego pliku konfiguracyjnego – blog.conf

```
# set the full URL to your weblog (required for absolute links).
# e.g. BLOG_URL="http://weblog.user/~foo" ("/" gets appended automatically)
BLOG_URL="http://copyme.sfery.org"

# stylesheet used for your weblog (should always be relative link).
BLOG_CSS="styles/nb_steebleblue.css"

# set default editor for your weblog (defaults to $EDITOR).
NB_EDITOR="$EDITOR"

# title of your weblog.
BLOG_TITLE="Mój blog"
```

7. Dokumentacja znajduje się w archiwum z NanoBloggerem w katalogu: docs/nanoblogger.html oraz pod adresem: http://nanoblogger.sourceforge.net

8. W NanoBloggerze nie istnieje rozróżnienie na kategorie i tagi. Tag oznacza tak naprawdę kategorię.

9. http://nanoblogger.sourceforge.net/downloads/nanoblogger-extra-3.4.2.tar.gz

10. Zakładając, że CSS i HTML nie stanowią dla czytelnika problemu.

Czytelnik zainteresowany jest na przykład zaopatrzeniem swojego bloga w system komentarzy, zapraszam do zapoznania się z dokumentacją.

### Tidy

Wtyczka ta pozwala na zintegrowanie naszego blogu z popularnym konsolowym validatorem HTML. Dzięki temu mamy możliwość odpluskowania kodu naszych wpisów. Wszelkie odnalezione problemy z naszym kodem zapisywane są w pliku `Tidy.log`, który znajdziemy w katalogu bloga. Wtyczka do działania wymaga zainstalowania w systemie validatora Tidy. Ustawienia wtyczki możemy zmieniać w pliku `blog.conf`.

### Markdown

Osoby, którym nie uśmiecha się redagowanie swoich wpisów, w HTML-u na pewno ucieszą się, że dzięki wtyczce Markdown mamy możliwość pisania w tym bardzo prostym języku. Wtyczka wymaga zmiany formatu wpisu `z raw na markdown` oraz zainstalowania w systemie pakietu `python-markdown`. Ustawienia wtyczki możemy zmieniać w pliku `blog.conf`.

### Moods – buźki

Wtyczka ta umożliwia po prostu zamianę grupy znaków na tzw. buźki. Wymaga skopiowania do katalogu bloga katalogu `moods`, który znajduje się w głównym katalogu NanoBloggera. Aby buźki zostały wyświetlane, musimy w polu format wpisu dodać po przecinku `wpis moods`. Lista grup znaków zamienianych na buźki znajduje się w pliku `moods/moods.conf`.

## Dodatek A

### Automatyczna synchronizacja

Ciągłe przerzucanie plików na serwer WWW może stać się po pewnym czasie dość męczące. Na szczęście NanoBlogger pozwala w prosty sposób zautomatyzować tę czynność. Opisany tu sposób odnosi się do protokołu FTP, jednak nic nie stoi na przeszkodzie, by przy odrobinie wysiłku skonfigurować automatyczną synchronizację z wykorzystaniem innego protokołu.

Aby dodać wsparcie dla automatycznej synchronizacji, musimy w pliku `blog.conf` zmodyfikować wpis `BLOG_PUBLISH_CMD="ftp <serwer>"` oraz utworzyć w naszym katalogu domowym plik `.netrc` i wyedytować zgodnie z Listingiem 6.

### Podsumowanie

Nanoblogger dzięki wykorzystaniu Basha oraz dobrej dość dobrej dokumentacji jest CMS-em prostym, elastycznym, łatwym w rozbudowie, oraz modyfikacji. Wszystko to sprawia, że przy naprawdzie niewielkim nakładzie pracy oraz mocy obliczeniowej serwera możemy prowadzić w wygodny, a przede wszystkim tani sposób niewielkiego bloga. Polecam NanoBloggera wszystkim

### **Listing 5.** Lista komend umożliwiających wykonanie popularnych czynności

```
#Dodawanie wpisów|kategorií|artykułów|blogu
nb -b /<ścieżka>/<do>/<bloga>/ add [entry|tag|article|weblog]

#Wyświetlenie listy wpisów
nb -b /<ścieżka>/<do>/<bloga>/ -l

#Edycja wpisu
nb -b /<ścieżka>/<do>/<bloga>/ edit entry <ID wpisu>

#Kasowanie wpisów
nb -b /<ścieżka>/<do>/<bloga>/ -d <ID wpisu>

#Aktualizacja np. po ręcznej zmianie plików
nb -b /<ścieżka>/<do>/<bloga>/ update [all|tag|DATE|articles|feeds>tag <ID wpisu>]
```

### **Listing 6.** Przykładowa zawartość pliku `.netrc`

```
machine <serwer> login <login> password <hasło>
macdef init
passive on
prompt off
cd /<zdalna>/</ścieżka>/<do>/<blogu>/
lcd /<lokalna>/</ścieżka>/<do>/<blogu>/
mput *.*
cd archives
lcd archives
mput *
quit
```

### **Listing 4.** Zmiana konfiguracji

```
nb -b /<ścieżka>/<do>/<bloga>/ configure update all
```

niebojącym się konsoli osobom, które chcieliby od czasu do czasu skrobiąć notkę na swojego bloga.

Na zakończenie tej części artykułu „” pozostaje mi tylko zaprosić drogich Czytelników do części drugiej. W następnej części przyjrzymy się z bliska kolejnemu statycznemu generatorowi treści – Jekyll.

W razie jakichkolwiek pytań oraz uwag dotyczących tej publikacji zapraszam do skontaktowania się z autorem poprzez adres e-mail podany w ramce „O autorze”.

## KACPER PLUTA

*Na co dzień pracuje jako administrator oraz studiuje w Wyższej Szkole Informatyki w Łodzi. Pasjonat Wolnego Oprogramowania, dobrej muzyki i książek. Od niedawna stara się blogować na <http://blog.kofeina.it> Wszelkie pytania do artykułów proszę kierować na [kacper.pluta@kofeina.it](mailto:kacper.pluta@kofeina.it)"*

# Certyfikaty AIRTrust

## Gwarancja autentyczności Twoich aplikacji

Podpisując swoje aplikacje certyfikatem Adobe AIR, udowadniasz swoją tożsamość użytkownikom, co przekłada się bezpośrednio na wzrost zaufania do Twojego produktu i Twojej firmy, jednocześnie pokazujesz użytkownikom, że Twoje oprogramowanie można bezpiecznie zainstalować i używać.

### Certyfikat AIR – co to takiego i do czego służy?

Kiedy Twoja aplikacja jest już gotowa, aby dostarczyć ją klientom, nadeszła pora, by podpisać ją za pomocą certyfikatu aby instalator Adobe AIR mógł ją poprawnie zainstalować w systemie użytkownika. Środowisko uruchomieniowe Adobe AIR działa z takimi samymi uprawnieniami jak każda natywna aplikacja, zezwalając na dostęp do lokalnego dysku twardego, sieci itp. Wymagając, aby Twój kod był podpisany, Adobe zapewnia Twoim klientom pewien poziom bezpieczeństwa po przez uwiarygodnienie źródła pochodzenia aplikacji.

Certyfikat AIR jak każdy certyfikat służący do cyfrowego podpisywania kodu programu (*Code Signing Certificate*), jest tworem bardzo zbliżonym do certyfikatu SSL. Składa się z części publicznej (podpisu widniejącego w aplikacji) oraz prywatnej – klucza niezbędnego do podpisania aplikacji. Ponieważ certyfikat Adobe AIR musi być wydany przez Zaufany Ośrodek Certyfikacji

(*Trusted Certificate Authority*), dobrze jest zakupić certyfikat u zaufanych dostawców certyfikatów lub ich partnerów. Do najbardziej rozpoznawalnych marek należą: GlobalSign, Thawte czy VeriSign.

Certyfikaty Adobe AIR Code Signing pozwalają na podpisanie wszelkiego rodzaju aplikacji Adobe AIR z poziomu takich środowisk jak Flex Builder, Flex SDK, Flash czy Dreamweaver. Kiedy podpisujesz swój program za pomocą zaufanego certyfikatu, jego użytkownicy mają pewność, że pochodzi on z zaufanego źródła (od Ciebie) i nie był zmodyfikowany (np. przez złośliwy kod) od momentu, kiedy podpisałeś swoją aplikację certyfikatem Adobe AIR. Oznacza to również, że użytkownicy nie będą musieli więcej oglądać ostrzeżenia o nieznianym wydawcy aplikacji (Rysunek 1).

Zamiast tego, zostaną w jasny sposób powiadomieni, kto jest autorem aktualnie instalowanego programu (Rysunek 2).



Rysunek 1. Instalacja niepodpisanego programu



Rysunek 2. Instalacja podpisанego programu

Podpisując swoje aplikacje certyfikatem Adobe AIR:

- Udowadniasz swoją tożsamość użytkownikom, co przekłada się bezpośrednio na wzrost zaufania do Twojego produktu i Twojej firmy,
- Gwarantujesz, że oprogramowanie nie zostało zmodyfikowane np. przez cracker'a,
- Pokazujesz użytkownikom, że Twoje oprogramowanie można bezpiecznie zainstalować i używać,
- Chronisz swoją markę.

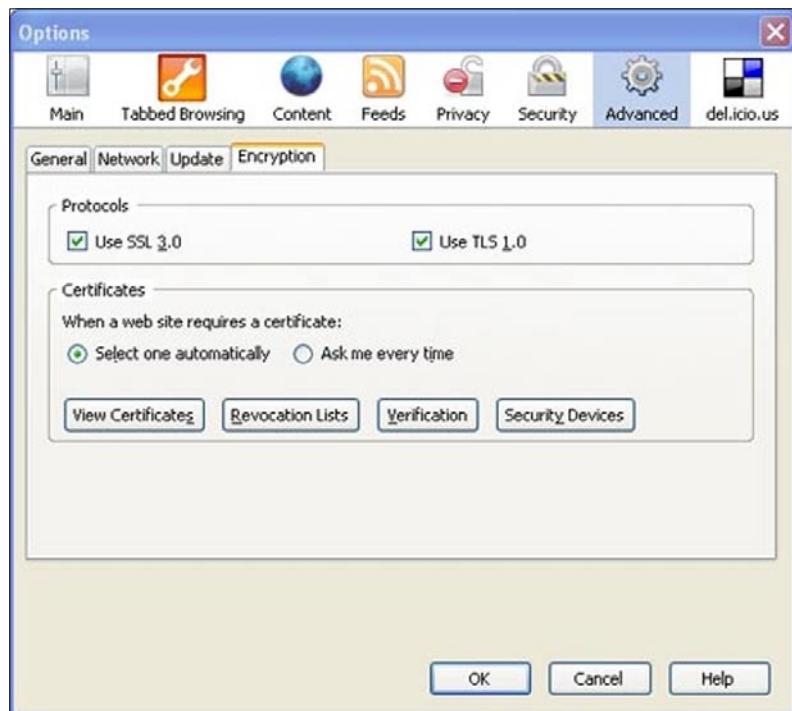
### Jak i gdzie kupić certyfikaty AIR?

Aby zamówić certyfikat AIR np. od GlobalSign, przede wszystkim należy podać dokładne dane swoje/firmy. Ważne, aby nie podawać żadnych skrótów ani nazw potocznych, dane firmy muszą być dokładnie tak jak jest podane w dokumentach firmy, gdyż inaczej weryfikacja zamawiającego będzie bardzo utrudniona i może zakończyć się niepowodzeniem. Wydawanie certyfikatu trwa do 10 dni roboczych. W tym czasie przyjmowane jest zamówienie, następnie dokonywana jest weryfikacja firmy/osoby. Jeśli z jakiś powodów wystawca certyfikatu nie będzie w stanie samodzielnie poprawnie zweryfikować firmy np. w publicznym rejestrze KRS, weryfikacja może zostać przeprowadzona na dwa sposoby:

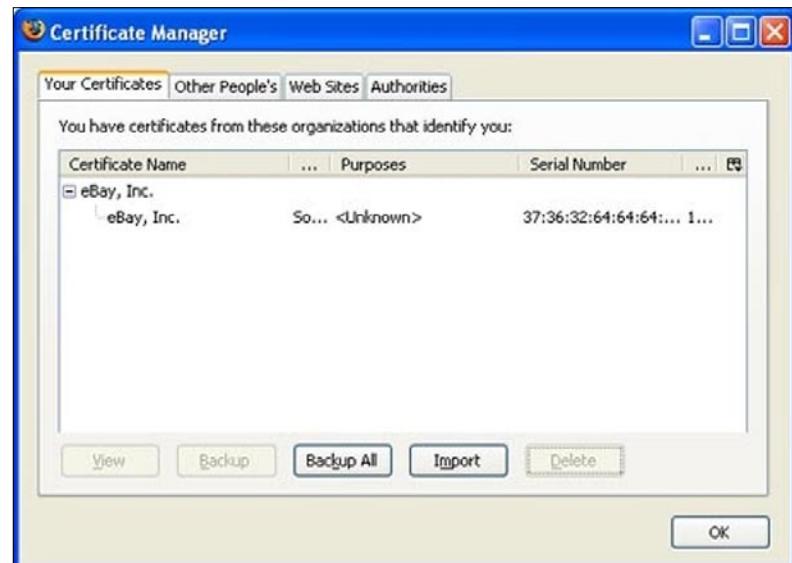
- Poprzez wysłanie przez GlobalSign listu z hasłem na adres widniejący w dokumentach rejestracji firmy. Po otrzymaniu tego listu należy zadzwonić pod podany w nim numer i podać zawarte w liście hasło.
- Jeśli firma posiada zarejestrowany telefon na firmę, w takim przypadku można wysłać skan ostatniej faktury telefonicznej (musi być widoczny dokładny numer telefonu oraz dokładna nazwa firmy – zgodna z nazwą firmy, na którą zamawiany jest certyfikat). Po przesłaniu faktury GlobalSign zadzwoni pod numer widoczny na fakturze w celu weryfikacji firmy.

Po weryfikacji firmy certyfikat zostanie wydany w ciągu jednego dnia roboczego, a następnie zostanie przesłany klientowi.

Zakupu certyfikatu można dokonać bezpośrednio na stronie dostawcy certyfikatu (opcja z reguły droższa) lub poprzez partnerów dostawców (zwykle jest to tańsze roz-



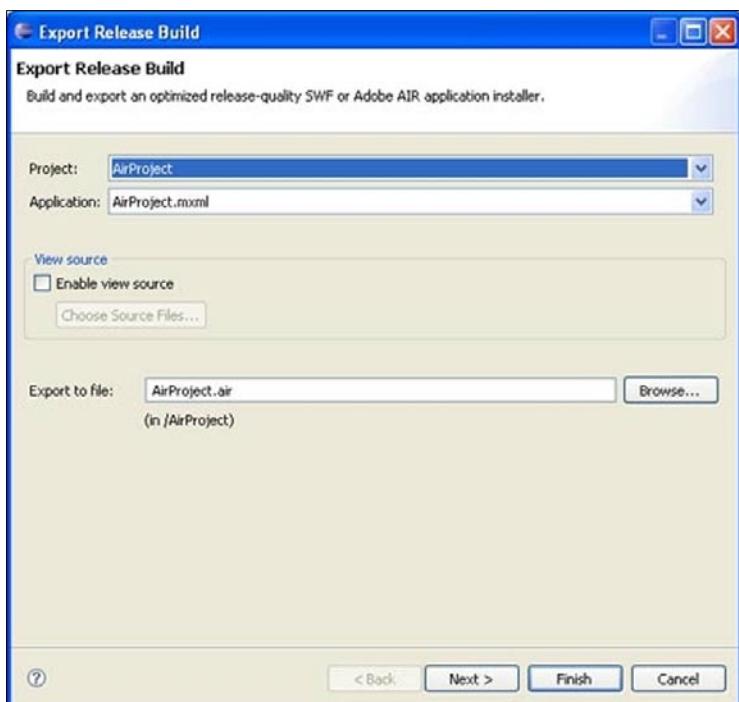
Rysunek 3. Firefox – Ustawienia szyfrowania



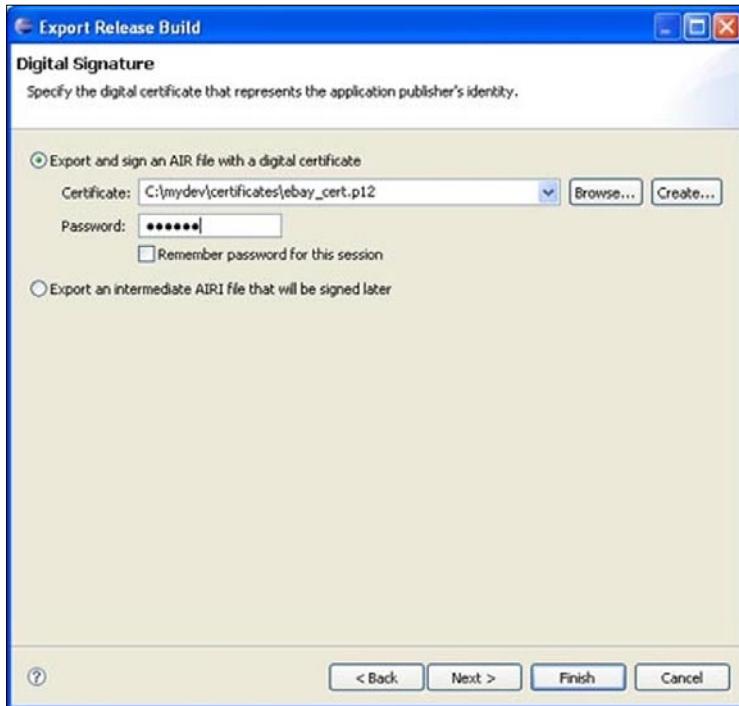
Rysunek 4. Manager certyfikatów przeglądarki Firefox



Rysunek 5. Tworzenie hasła do certyfikatu



Rysunek 6. Kreator wydania aplikacji



Rysunek 7. Kreator wydania aplikacji – podpisywanie kodu

wiązanie). W Polsce certyfikaty AIR można nabyć np. przez stronę <http://www.airtrust.pl>

## Instalacja certyfikatu AIR

Kiedy już nabędziesz certyfikat i zainstalujesz go ze strony dostawcy (z reguły sprowadza się to do kliknięcia linku „Instaluj mój certyfikat” w mailu od wystawcy certyfikatu). Następny krok to eksport certyfikatu z przeglądarki, której użyliśmy do instalacji do pliku. Pozwoli to użyć certyfikatu do podpisywania aplikacji, jak i zapewni możliwość wykonania kopii zapasowej.

Aby wyeksportować certyfikat z przeglądarki Firefox, należy wybrać *Opcje* z menu *Narzędzia*, następnie wybrać zakładkę *Szyfrowanie* z sekcji *Zaawansowane*.

Następnie trzeba kliknąć „Przeglądaj certyfikaty” i przejść do zakładki „Certyfikaty użytkownika” (Rysunek 3., Rysunek 4).

Wybierz certyfikat, który chcesz wyeksportować, i kliknij przycisk „Kopia zapasowa”. Firefox zapisze Twój certyfikat oraz jego klucz prywatny w pliku P12 (format PKCS12), który to będzie potrzebny do podpisywania aplikacji AIR z poziomu środowisk takich jak Flexbuilder czy Dreamweaver. Dla niektórych środowisk, konieczna będzie zmiana rozszerzenia zapisanego pliku na .pfx.

Po podaniu ścieżki, gdzie chcesz zapisać swój certyfikat, kreator poprosi o utworzenie hasła do certyfikatu – jest to ważne, gdyż dzięki temu nikt niepowołany, nawet jeśli ukradnie Twój certyfikat, nie będzie w stanie z niego skorzystać (Rysunek 5).

Po kliknięciu OK, Twój certyfikat będzie gotowy do podpisywania aplikacji AIR.

## Podpisywanie aplikacji AIR z wykorzystaniem Flex Buildera

Podpisywanie aplikacji AIR z poziomu Flex Buildera jest prostym procesem. Kiedykolwiek eksportujesz aplikację AIR z Flex Buildera 3, podpisywanie kodu będzie integralnym krokiem całego procesu. Nie można wyeksportować aplikacji bez podpisania jej chociażby samodzielnie podpisany testowym certyfikatem, jaki można utworzyć w tym samym kroku kreatora. Aby rozpocząć eksport aplikacji, wybierz z menu *Select Project* > *Export Release build*. Powinieneś zobaczyć dialog jak na Rysunku 6.

Wypełnij pola formularza tak jak do tej pory i kliknij *Next*.

W polu *Certificate* wskaż zapisany plik z certyfikatem, który wyeksportowałeś wcześniej z Firefoxa. Wprowadź hasło, które podałeś, eksportując certyfikat, i kliknij – *Finish*. Powinieneś teraztrzymać plik .air zawierający Twoją podpisaną aplikację. Będzie się on znajdował w katalogu Twojego projektu.

Nadszedł teraz czas na instalację aplikacji, tak jak zrobiłby to normalny użytkownik, który właśnie ściągnął Twój program z Internetu. Jeśli wszystko przebiegnie prawidłowo, Twoja aplikacja jest gotowa do publikacji w sieci.

## SEBASTIAN FYDA

*Autor jest starszym administratorem systemów operacyjnych w Janmedia Interactive. Kontakt z autorem: sfyda@janmedia.pl*



Maciąc na uwadze ciągły rozwój systemów bezpieczeństwa informacji za główny cel postawiliśmy sobie dostarczenie Państwu najnowszych i najlepszych praktyk z obszaru zarządzania bezpieczeństwem IT. Robimy to w sposób jasny, przystępny, opierając się o najnowsze normy typu PN-ISO/IEC 27001:2007, PN-ISO/IEC 17799:2007. Wieloletnie doświadczenie i prezentowany poziom merytoryczny stawia nas bezwarunkowo jako liderów na rynku polskim w obszarach szkoleń i certyfikacji bezpieczeństwa.

- **Infrastruktura klucza publicznego i podpis elektroniczny w praktyce**  
22–24 listopada 2010
- **BCM – Zarządzanie Ciągłością Działania**  
25 listopada 2010
- **Szkolenie przygotowujące do egzaminu CISSP (Certyfikat Certified Information Systems Security Professional)**  
29 listopada – 3 grudnia 2010
- **Forum Ochrony Danych Osobowych**  
grudzień 2010

Kontakt i szczegóły:  
**Edyta Szewc**  
tel. +48 22 427 36 70 | fax +48 22 244 24 59  
[edyta.szewc@software.com.pl](mailto:edyta.szewc@software.com.pl)



Przyjdź i zdobądź:  
**Generalny Certyfikat Inżyniera Bezpieczeństwa IsecMan**

# Biblioteka Qt

## Profesjonalny framework dla każdego

Istnieje wiele różnic pomiędzy amatorskim programowaniem a profesjonalnym tworzeniem systemów informatycznych. Oprócz oczywistych, takich jak ogromna różnica w złożoności kodu i liczby osób zaangażowanych w projekt, jedną z najważniejszych jest zastosowanie odpowiednich bibliotek i frameworków programistycznych. Dostarczają one klas do obsługi tak podstawowych technologii, jak komunikacja w protokole TCP/IP, dokumentów XML, graficznych interfejsów użytkownika oraz programowania zgodnego ze wzorcem model-widok-kontroler (MVC – Model-View-Controller). Jedną z najlepszych i najszerzej stosowanych bibliotek programistycznych jest Qt. Ten krótki artykuł stanowi pilota nowego cyklu poświęconego programowaniu z wykorzystaniem tej biblioteki.

**J**ednym z największych problemów, jakie napotykają programiści, którzy chcą zacząć pisać rozbudowane aplikacje, jest wybór odpowiedniej biblioteki programistycznej. Podstawowe biblioteki większości współczesnych języków programowania wysokiego poziomu często okażą się niewystarczające. Oczywiście możemy np. własnoręcznie zaimplementować obsługę standardu DOM, służącego do mapowania zawartości dokumentu XML do struktury drzewa, jednak mija się to z celem. Jeżeli nie chcemy powtórnie wynajdywać koła, powinniśmy zapoznać się z możliwościami popularnych bibliotek programistycznych i wybrać najbardziej odpowiadającą naszym potrzebom.

### Qt – framework dla każdego

Qt jest na chwilę obecną jedną z najczęściej wykorzystywanych bibliotek programistycznych dla języków C++, Java, Python oraz wielu innych. Biblioteka ta jest jednak najczęściej kojarzona z językiem C++ ze względu na fakt, iż jest to język, w którym została ona napisana.

Oprócz dostępności bindingów dla niemal wszystkich popularnych języków programowania, Qt może poszczycić się prawdziwą wieloplatformowością. Do obsługiwanych architektur sprzętowo-programowych należą: Linux, Microsoft Windows, Mac OS X, Linux w systemach wbudowanych (PDA, smart-

fony), Symbian, Microsoft Windows CE (minimalistyczna wersja systemu Windows, przeznaczona dla urządzeń wbudowanych) oraz Maemo. Tak dłuża lista robi naprawdę dobre wrażenie, szczególnie jeżeli weźmiemy pod uwagę fakt, iż komplikacja dla odmiennych architektur odbywa się zazwyczaj z jednego kodu źródłowego. Pozwala to na ogólną oszczędność czasu przy opracowywaniu aplikacji wieloplatformowych.

### Qt Creator – środowisko programistyczne

Jednym z problemów napotykanych przez początkujących programistów jest wybór odpowiedniego środowiska programistycznego (IDE – *Integrated Development Environment*). W trakcie pobierania biblioteki Qt możemy zdecydować się naściągnięcie IDE Qt Creator pozwalającego na szybkie rozpoczęcie programowania przy jej wykorzystaniu (Rysunek 1).

Qt Creator należy do najlepszych wieloplatformowych środowisk programistycznych. Doskonale nadaje się zwłaszcza dla początkujących użytkowników biblioteki Qt, a to za sprawą łatwego dostępu do dokumentacji poszczególnych klas i modułów. Gorąco polecam to środowisko programistyczne każdemu, kto dopiero rozpoczyna pracę z biblioteką Qt. Na szczególną uwagę zasługują również doskonale udokumentowane przykłady, prezentujące możliwości poszczególnych modułów biblioteki.

## Moduły biblioteki Qt

W celu lepszej organizacji, klasy składające się na bibliotekę Qt podzielone zostały na moduły. Najważniejsze z nich to:

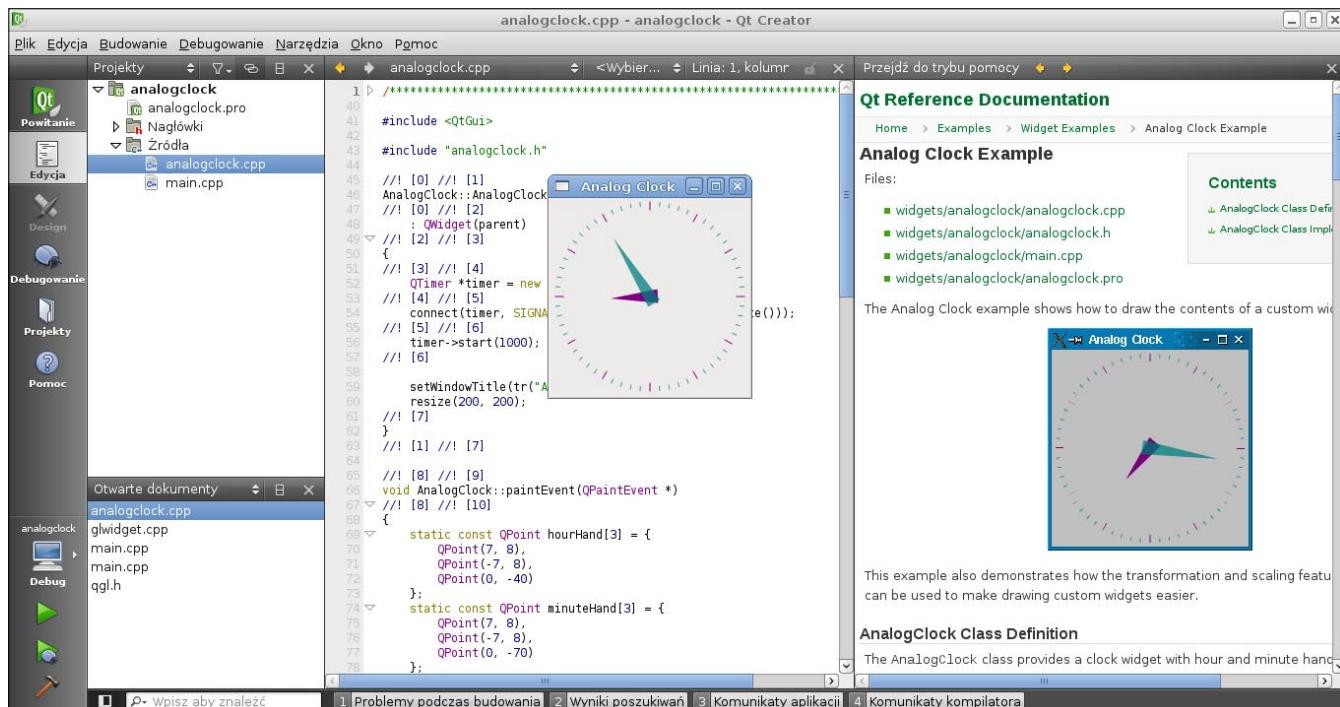
- **QtCore** – moduł zawierający podstawowe klasy, wykorzystywane przez inne moduły, takie jak np. `QString` (klasa reprezentująca串 znaków), `QList` (klasa reprezentująca listę) oraz `QThread` (klasa reprezentująca wątek).
- **QtGui** – moduł zawierający klasy służące do budowania graficznych interfejsów użytkownika, takie jak np. `QPushButton` (klasa reprezentująca przycisk), `QComboBox` (klasa reprezentująca listę rozwińalną) oraz `QTreeView` (klasa reprezentująca widok drzewa).
- **QtNetwork** – moduł zawierający klasy służące do budowania aplikacji sieciowych, takie jak np. `QTcpSocket` (klasa reprezentująca socket korzystający z protokołu TCP na poziomie warstwy transportu), `QUdpSocket` (socket korzystający z protokołu UDP), `QFtp` (klasa reprezentująca implementację protokołu FTP po stronie klienta) oraz `QSSlSocket`

(gniazdo sieciowe korzystające z protokołu kryptograficznego SSL).

- **QtSql** – moduł zawierający klasy służące do budowania aplikacji korzystających z baz danych SQL, takie jak np. `QSqlDatabase` (klasa reprezentująca połączenie z bazą danych), `QSqlTableModel` (klasa reprezentująca model pojedynczej tabeli bazy danych) oraz `QSqlRecord` (klasa enkapsulująca rekord bazy danych).
- **QtXML** – moduł zawierający klasy służące do budowania aplikacji obsługujących dane zawarte w plikach XML, takie jak np. `QDomDocument` (klasa reprezentująca dokument DOM), `QDomNode` (klasa reprezentująca pojedynczy węzeł w drzewie DOM) oraz `QXmlReader` (klasa stanowiąca interfejs dla parserów XML).
- **QtOpenGL** – moduł zawierający klasy służące do budowania aplikacji korzystających z biblioteki OpenGL, takie jak np. `QGLWidget` (klasa reprezentująca widżet renderujący grafikę), `QGLShader` (klasa reprezentująca shadery napisane w językach GLSL oraz GLSL/ES) oraz `QGLFormat` (klasa reprezentująca format wyświetlania kontekstu renderowania OpenGL).

**Tabela 1.** Licencjonowanie biblioteki Qt

	Licencja LGPL	Licencja komercyjna
Za licencję pobierana jest opłata	Nie	Tak
Zmiany w kodzie biblioteki muszą być upublicznie	Tak	Nie
Możliwość tworzenia własnościowych aplikacji	Tak	Tak
Pomoc techniczna	Tak (konieczność wykupienia)	Tak (pełna)



**Rysunek 1.** Środowisko programistyczne Qt Creator

Nie są to oczywiście wszystkie moduły zawarte w bibliotece Qt. W najnowszej wersji, oznaczonej numerem 4.7, dostępne są ponadto moduły: `QtSvg` (obsługa formatu SVG), `QtOpenVG` (obsługa API OpenVG, służącego do obsługi akcelerowanej dwuwymiarowej grafiki wektorowej w systemach wbudowanych), `QtXmlPatterns` (obsługa technologii związanych z XML: XML Schema, XPath, XQuery oraz XSLT), Phonon (framework multimedialny) oraz QtTest (framework dla testów jednostkowych).

Jak widać, biblioteka Qt dostarcza klas pozwalających na wygodne zastosowanie zaawansowanych technologii. Opisem praktycznego wykorzysta-

nia dużej części z nich zajmiemy się w cyklu, którego pilotem jest ten krótki artykuł.

### Twoje prawa, czyli licencja

Biblioteka Qt wraz z narzędziami dodatkowymi dostępna jest na zasadzie podwójnego licencjonowania: licencji komercyjnej oraz LGPL. Porównanie pomiędzy licencjami zawarte zostało w ramce Licencjonowanie biblioteki Qt. Oczywiście jest, iż większość Czytelników nie zdecyduje się na wykupienie licencji komercyjnej, tym bardziej więc cieszą liczne prawa wynikające z liberalnej licencji LGPL.

### Qt w systemach wbudowanych

Jak już napisałem, Qt jest biblioteką wieloplatformową. Przy jej pomocy możemy tworzyć aplikacje, które będą działały nie tylko pod kontrolą systemów operacyjnych dla komputerów PC, lecz również na urządzeniach mobilnych.

Dwa z artykułów cyklu zostaną w pełni poświęcone programowaniu dla urządzeń działających pod kontrolą systemu operacyjnego Symbian. Zajmiemy się również programowaniem dla Linuksa w systemach wbudowanych. Obydwa te zadania wymagają użycia dodatkowych narzędzi pozwalających na testowanie aplikacji bez konieczności posiadania rzeczywistych urządzeń. Zapoznamy się między innymi z symulatorem Qt Simulator (Rysunek 2) oraz wirtualnym buforem ramki (Virtual Framebuffer) wykorzystywanym do testowania aplikacji tworzonych dla systemu Linux bez X-ów.

### Kiedy startujemy?

Pierwszy artykuł cyklu pojawi się w numerze, który ukaże się 30 listopada br. Zaczniemy od wprowadzenia do biblioteki Qt – dowiesz się, jakie są zależności pomiędzy najczęściej wykorzystywany klasami oraz jak tworzyć interfejsy graficzne – zarówno przy użyciu graficznego narzędzia Qt Designer, jak i bezpośrednio w kodzie aplikacji. Poznasz praktyki, które warto stosować tworząc aplikacje korzystające z biblioteki Qt oraz dowiesz się, czego za wszelką cenę powinieneś unikać.

Jeżeli masz pytania odnośnie zawartości cyklu, nie wahaj się napisać e-maila – mój adres znajdziesz w rubryce na końcu artykułu. Do usłyszenia!

---

### RAFAŁ KUŁAGA

*Autor interesuje się bezpieczeństwem systemów informatycznych, programowaniem, elektroniką, muzyką rockową, architekturą mikroprocesorów oraz zastosowaniem Linuksa w systemach wbudowanych.*

*Kontakt z autorem: rl.kulaga@gmail.com*



Rysunek 2. Testowanie aplikacji dla urządzenia mobilnego opartego na systemie Symbian w symulatorze Qt Simulator



HEUTHES

### Systemy bankowe, ISOF

HEUTHES istnieje na rynku od 1989 r. Obok systemów informatycznych dla banków, oferuje nowoczesne oprogramowanie do obsługi firm. System ISOF jest udostępniany klientom w trybie SaaS lub licencji. Pracuje na platformie Linux i zawiera m.in. takie moduły jak CRM, DMS, Magazyn, Sprzedaż, Logistyka oraz Rachunkowość.

<http://www.isof.pl>

**Bet@Soft®**  
Sp. z o.o.

### BetaSoft Sp. z o.o.

Jednym z naszych autorskich rozwiązań jest system eDokumenty realizujący wymagania DMS i CRM. Pracuje on na platformach Linux oraz Windows i należy do najnowocześniejszych aplikacji webowych działających w technologii AJAX. System jest zintegrowany z OpenOffice, MS Office i programami ERP.

[www.betasoft.pl](http://www.betasoft.pl),  
[www.edokumenty.eu](http://www.edokumenty.eu)

**TTS** Company

### TTS Company Sp. z o.o.

Sprzedaż i dystrybucja oprogramowania komputerowego. Import programów na zamówienie. Ponad 200 producentów w standardowej ofercie. Chcesz kupić oprogramowanie i nie możesz znaleźć polskiego dostawcy? Skontaktuj się z nami – sprowadzimy nawet pojedyncze licencje.

[www.OprogramowanieKomputerowe.pl](http://www.OprogramowanieKomputerowe.pl)



### Wyższa Szkoła Informatyki

Informatyka, Ekonomia, Fizjoterapia (NOWOŚĆ), Pedagogika, Wychowanie Fizyczne, Artystyczna Grafika Komputerowa, Architektura Wewnętrz (NOWOŚĆ) Studia na odległość e-learning (Informatyka II stopnia) Kompleks sportowy w Łodzi (hala, basen, siłownia, sauna). Wydziały zamiejscowe: Włocławek, Bydgoszcz, Opatówek.

<http://www.wsinf.edu.pl>

## OFERTA SKIEROWANA DO FIRM

**Wyślij do nas: logo firmy, dane kontaktowe i informacje o firmie.  
Reklama przez 12 kolejnych numerów tylko za 600 PLN + VAT.**

**Skontaktuj się z nami:**

**[linux@software.com.pl](mailto:linux@software.com.pl) tel. 22 427 36 52**

# Za software jak za wodę

Zmiany na rynku spowodowane kryzysem gospodarczym postawiły nowe wyzwania przed sektorem IT. Firmy potrzebują nowych rozwiązań, by móc maksymalizować zyski z projektów, obniżając jednocześnie ryzyko. Era długoterminowych kontraktów powoli mija, dzisiaj przedsiębiorcy za kadrę, sprzęt i oprogramowanie IT chcą płacić tak, jak za wodę - według zużycia.

**M**odel SaaS (Software as a Service) i związana z nim technologia chmury nie są dla branży IT niczym nowym. - *The Cloud Computing to nie tylko przyszłość computingu. To jego teraźniejszość oraz cała przeszłość. W kategorii technologii to po prostu komputer podłączony do sieci, a w kategorii business model – to zwykły wynajem. To SAS ze zmienioną nazwą. Można to nazwać 'Cloud Computing', ale my to robimy już od dekady* - takimi słowami Larry Ellison, założyciel i CEO Oracle Corporation wyraził swój sceptyczny odnośnie kariery, jaką w ostatnich czasach robi owo zjawisko.

## Co nowego?

Rzeczywiście tak duże firmy podobne działania podejmowały już pięć czy dziesięć lat temu. Z drugiej strony trudno nie zauważyc znaczenia, jakie nowy trend ma dla małych i średnich przedsiębiorstw, a przede wszystkim tych, którzy dopiero zaczynają własną działalność. Nowy czy nie – SaaS i Cloud Computing pozwalają na drastyczne obniżenie kosztów i częściowo ryzyka wiążących się z rozwijaniem nowych firm: zamiast drogich, minimum 12-miesięcznych subskrypcji można wykorzystać model bliższy rachunkom za zużytą wodę lub doładowaniu kart telefonicznych. Oznacza to, że przedsiębiorca płaci tylko za to, co realnie używa. W przypadku serwerów płaci za ruch, a dodatkowo część firm oferujących taką usługę pozwala na dokonanie opłaty nie z góry, a pod koniec okresu rozliczeniowego. Takie rozwiązanie nie było możliwe jeszcze kilka lat temu, kiedy wynajem serwerów był drogi oraz mało elastyczny.

Wynajem oprogramowania ma też swoje zalety: jego aktualizacji zwykle podejmuje się dostawca, bez pobierania dodatkowych opłat. Nie traci się przy tym czasu na wdrażanie oprogramowania, nie trzeba szkolić administratorów czy kupować sprzętu. Wystarczy login, hasło oraz krótkie zapoznanie się z systemem. Zapewnione jest także bezpieczne przechowywanie danych oraz szkolenia w systemie e-learning. Z danych Internatio-

nal Data Corporation wynika, że taki sposób pozyskiwania oprogramowania pozwala zredukować koszty związane z informatyką o 30 – 70 procent. Wynajęcie oprogramowania minimalizuje też ryzyko związane z dużymi inwestycjami w software. - *Zdarzają się przypadki, w których ogłasza się przetarg na oprogramowanie, które jest niekompatybilne z obecnym systemem operacyjnym instytucji. Po zakupie okazuje się, że trzeba ogłosić kolejny przetarg, tym razem na sam system. To rodzi dodatkowe, niepotrzebne koszta* - wyjaśnia Marceli Smela, dyrektor generalny firmy IT Kontrakt, lidera wśród firm outsourcingowych oferujących specjalistów z branżą IT w Polsce. - *Decydując się na zewnętrznych specjalistów, mamy gwarancję, że taki scenariusz nie będzie miał miejsca. Dlatego bardzo często najlepszym rozwiązaniem nie jest klasyczny zakup oprogramowania, które i tak po jakimś czasie przestanie być aktualne. Duże oszczędności można wygenerować, wypożyczając pożyczane przez nas aplikacje w modelu SaaS. W takim przypadku klient płaci jedynie za zdalny dostęp do wybranego programu* – dodaje.

Najlepszym dowodem na intensyfikację tego trendu jest nowa oferta Microsoftu, która opiera się jednocześnie na rozwiązaniu SaaS i korzysta z technologii chmury. Przedsiębiorcy, którzy zarejestrują się do testowania wersji beta, mogą skorzystać z oferty, która daje dostęp do pełnego pakietu usług Office 365. Płacąc 6 dolarów od użytkownika, firmy mogą korzystać z platformy, która została określona mianem Facebooka dla biznesu.

- *Wynajem aplikacji zamiast jej tworzenia lub kupna jest dzisiaj najlepszą opcją* – wyjaśnia Krzysztof Czerpak, key account manager firmy oferującej wynajem aplikacji In Flavo, czyli sklepu internetowego na Facebooku – *Dostawcy biorą na siebie odpowiedzialność za bezpieczeństwo danych, serwery, a także za aktualizację oprogramowania. Pozwala to przedsiębiorcy skupić się na prowadzeniu biznesu, na przykład sprzedaży w Internetie, bez martwienia się o to, jak jego sklep będzie wyglądał na monitorach o różnej rozdzielczości i na różnych*

przeglądarkach, czy jest funkcjonalny i jak go aktualizować. W razie, jeśli pomysł na biznes się nie sprawdzi, nie traci pieniędzy, które musiałby zainwestować w stworzenie lub kupno programu.

### Szukając oszczędności

Kryzys w światowej gospodarce to jeden z głównych czynników, które wpłynęły na popularność tego typu rozwiązań, zarówno w Polsce, jak i na całym świecie. Firmy szukają oszczędności, jednocześnie dążąc do ciągłego ulepszania swoich usług, dla podtrzymywania konkurencyjności oferty.

Zastosowanie Cloud Computing oprócz dostosowania do bieżących potrzeb firmy pozwala na obniżenie miesięcznych kosztów zużycia energii potrzebnej np. do utrzymania i chłodzenia maszyn. Outsourcing kadry IT pod względem zalet i generowanych oszczędności przypomina model SaaS. Dzięki temu rozwiązaniu firmy nie muszą inwestować w nowych pracowników – oszczędzają czas i pieniądze, jakie trzeba wyłożyć na długi i często skomplikowany proces rekrutacji oraz zapewnianie ciągłego rozwoju. Ponadto, jak zauważa Marcelli Smela, wiele przedsiębiorstw nie potrzebuje etatowych informatyków - Obsługa informatyczna w wielu firmach w większości przypadków ogranicza się do instalacji oprogramowania, szkoleń czy drobnych naprawach sprzętowych. Wszystkie te czynności nie są wykonywane stale, więc nie ma większego sensu zatrudniać informatyków na stałe. Wygodniej i taniej jest zlecić wykonanie tej samej usługi zewnętrznemu specjalistie - wyjaśnia dyrektor IT Kontrakt. Co więcej, w przypadku firm nie działających w branży IT rekrutacja specjalistów może być o tyle trudna, że nie mają one odpowiednich narzędzi do ocenienia swoich potrzeb w zakresie technologicznym oraz do zweryfikowania kompetencji potencjalnych pracowników. Jednak i w tym wypadku rynek wypracował metodę obchodzenia tych utrudnień. Tak samo, jak w przypadku oprogramowania lub sprzętu – można skorzystać z pomocy pośredników.

Model SaaS, outsourcing sprzętu oraz kadry IT to rozwiązanie optymalne, pozwalające na zachowanie, a czasem nawet podniesienie jakości, przy jednoczesnej racjonalizacji kosztów. Outsourcing wszedł w zupełnie nową fazę, stając się odpowiedzią na potrzeby przedsiębiorców. - W finansowo trudnych czasach klienci chcą więcej za mniej, co wyjaśnia zwrot w kierunku modelu SaaS – mówi Peter Galvin, dyrektor ds. marketingu amerykańskiej firmy ProofPoint – Nawet ci, którzy odnosili się cynicznie do możliwości oferowanych przez SaaS, przekonali się do tej idei, gdy zobaczyli zmniejszenie kosztów przy przejściu na model Software as a Service – dodaje Galvin.

---

MAGDALENA KOŁODZIEJ

Ściągnij nowy  
numer Hakin9  
11/2010!

HAKIN9 11/2010 ŚCIĄGNIJ ZA DARMO!

# HAKIN9

JAK SIĘ OBRONIĆ HARD CORE IT SECURITY MAGAZINE

nr 11/2010 (66)



Kieszonkowe security

URZĄDZENIA MOBILNE Z NICH TEŻ STRACISZ SWOJE CENNE DANE

DROGA PRZEZ SAD Z JABŁKAMI O INNYM PODĘJŚCIU DO BEZPIECZENSTWA URZĄDZEŃ MOBILNYCH FIRMY APPLE

NIEKONWENCJONALNE ATAKI WI-FI JAK WYKORZYSTAĆ LUKI BEZPIECZENSTWA W STANDARDZIE 802.11

PLUS SNORT, SOURCEFIRE, IPS WYWIAD Z BARTOSZEM ŚWIDERSKIM, LIDEREM OBSZARU STRATEGICZNEGO SECURITY W WORLDIT SYSTEMS

POBIERZ

# Jak przekonać kobietę do Linuksa?

Informatyka to dziedzina w dużej mierze opanowana przez mężczyzn. Owszem, są kobiety, które interesują się tą dziedziną, jednak stanowią one mniejszość. Niniejszy artykuł nie dotyczy zatem tych pań, które doskonale radzą sobie z różnym oprogramowaniem – one same doskonale wiedzą, co jest dla nich najlepsze. Ten artykuł dotyczy kobiet, które nie mają zielonego pojęcia o informatyce, korzystają z oprogramowania najbardziej znanego, ponieważ wydaje się ono najprostszym. Ale czy na pewno tak jest?

**M**oże mężczyźni nie do końca potrafią wytlumaczyć na czym polega „wspaniałość” Linuksa. Podsumowując w przewrotny sposób wstęp: ta publikacja tak naprawdę dotyczy mężczyzn! Znajdą się tu praktyczne wskazówki, jak przekonać kobietę, aby używała Linuksa.

Po pierwsze i najważniejsze: liczy się wygląd! Czy widzieliście kiedyś kobietę, która wchodząc do sklepu, bierze pierwsze ubranie z brzegu i je kupuje? Uważam, że jest to niemożliwe. Mężczyźni mówiąc o zaletach Linuksa, praktycznie w ogóle nie mówią o wyglądzie. Owszem wspominają coś o „prostym interfejsie”, ale co to tak naprawdę znaczy? Zamiast wyraźnie powiedzieć „Patrz, jakie są tu ładne ikony, efekty! I jak można zagospodarować miejsce na pulpicie”. Kobieta na pewno się zainteresuje. „Hmm... to brzmi interesująco, mów dalej”. Wtedy można pokazać np. <http://kde-look.org> i jesteśmy w niebie – można przemeblowywać i przemalowywać swój pulpit w zależności od dnia



tygodnia, od humoru i czegokolwiek kobieta zapragnie. Mając Linuksa i znając wyżej podaną stronę, będzie się czuła jak w prawdziwym butiku.

Kolejna ważna rada, która przyda się mężczyźnie namawiającego kobietę na zmianę oprogramowania... Porządek! Nie można ukryć w żaden sposób faktu, iż dzięki Linuksowi panuje w naszym komputerze idealny ład. Wszystko jest poukładane logicznie w folderach. Łatwo można odnaleźć to, czego szukamy. Oczywiście nie należy też popadać w skrajność i wymagać, aby system operacyjny sam utrzymywał nam „czystość”. Czasem musimy usunąć coś, ale poza tym zawsze możemy mieć pewność, że wszystko jest na swoim miejscu. Przytoczenie tego faktu może być niezwykle istotne dla kobiety pedantki, wręcz może to być dla niej wybawienie od użerania się z „brudami” w innym oprogramowaniu.

Powyższe argumenty trafią do większości kobiet. Jednak nie każda kobieta patrzy tylko na wygląd i to, czy coś jest na swoim miejscu czy też w zdecydowanej nieodpowiedniej lokalizacji. Pamiętajmy też o bezpieczeństwie! Chyba nie ma ani jednej osoby na świecie, która uważałyby, iż Linux jest niebezpieczny. Nawet użytkownicy Windowsa zdają sobie sprawę, że jeśli chodzi o bezpieczeństwo w Sieci, to fiński wynalazek chroni nas całkowicie i stu procentowo. Co to oznacza w praktyce? Zero użerania się z programami anty-

wirusowymi. Nie musimy już myśleć o tym, jakiego wybrać, jak go zainstalować itp. Posiadając Linuksa, zamiast wyżej wymienionych „atrakcji” mamy zero stresu i więcej miejsca na dysku.

Nie można zapomnieć, iż kobiety kochają robić zakupy, lubią mieć dużo nowych i ładnych rzeczy, ale... tak naprawdę nie lubią, kiedy im ubywa pieniędzy z portfela. Chyba nie muszę już mówić, że kolejnym argumentem, który ma przekonać kobietę do zainstalowania na swym komputerze Linuksa, jest zerowa cena! Czyż to nie cudowne uczucie, kiedy ma się wszystko za darmo, począwszy od systemu operacyjnego po różne programy? Chyba nikt nie zdoła się oprzeć temu. No i oczywiście wszystko legalnie.

Konkluzja z artykułu jest taka – do kobiety trzeba prosto i rzeczowo, zwłaszcza jeśli opowiadamy o czymś, na czym dana pani się nie zna. Niewskazane, wręcz zabronione powinno być używanie w takich przypadkach skomplikowanych terminów, jak na przykład „terminal”. Jeżeli już bardzo tego pragniemy, to bądźmy przygotowani w dużej mierze na porażkę, gdyż zwyczajnie nasz odbiorca nie zrozumie nic z wypowiedzianych formułek.

---

**BEATA ROKICKA**

## Twój obecny admin

{ # Nie 3ędz!esz M! mówić !! }



**Hosting360**

[www.hosting360.pl](http://www.hosting360.pl)  
specjalistyczny hosting non stop

**Jasne! Chętnie pomogę...**



- Streaming
- Serwery dedykowane
- Serwery VPS
- Outsourcing IT
- Certyfikaty SSL

**Hosting360**

REKLAMA