Undervolting pod Linuksem

Sławomir Skrzyniarz

Przeraża cię myśl o tym, że już wkrótce otrzymasz rachunek za energię elektryczną? Boisz się kwoty jaką zobaczysz na fakturze za prąd? Może już masz dość słuchania bez przerwy szumiącego wentylatora na procesorze i buchającego gorąca z wnętrza obudowy? A może za krótko trzyma bateria w twoim laptopie? Jeśli tak, to chyba nastąpił właściwy moment, by pomyśleć nad włączeniem oszczędzania energii w twoim komputerze.

linux@software.com.pl

Na temat oszczędzania energii w Linuksie napisano już bardzo wiele artykułów. Wiele z nich opowiada o takich programach jak *cpufreq*, *cpudyn*, *powersaved*, *cpufreqd*, itp. Są to bardzo dobre programy, które pozwalają zmniejszyć prędkość procesora, gdy nie jest wykorzystywana ich cała moc obliczeniowa i dzięki temu zaoszczędzić dużo energii. Jest to podejście jak najbardziej prawidłowe. Co jednak mają zrobić użytkownicy, którzy muszą wykorzystywać całą dostępną moc obliczeniową? Muszą wybierać pomiędzy oszczędnością, a wydajnością. Niniejszy artykuł prezentuje podejście niestandardowe, które pozwoli zachować maksymalną wydajność i równocześnie zmniejszyć pobór mocy komputera nawet o przeszło trzydzieści procent. Mowa tutaj o tzw. *undervoltingu*, czyli obniżaniu napięcia rdzenia procesora. Operacja ta pozwala obniżyć drastycznie ilość pobieranej energii elektrycznej oraz wydzielanego ciepła. Znalezienie minimalnego napięcia rdzenia odbywa się eksperymentalnie i wymaga testów potwierdzających stabilność systemu na nowych ustawieniach. Dobrą wiadomością, jest to, że zaprezentowane poniżej rozwiązanie oparte o program CPU Power w większości dystrybucji nie wymaga ingerencji w jądro systemowe, a sprowadza się jedynie do pobrania źródeł programu z Internetu, skompilowania go i właściwego skonfigurowania.

Konfiguracja wstępna

Konfigurację należy rozpocząć od pobrania programu CPU Power ze strony *http://tuxamito.com.es/cpupw/*, a następnie z uprawnieniami administratora wykonywać wszystkie dalej opisane czynności. Program należy rozpakować poleceniem *tar -z*

Listing 1. Przykładowy skrypt programu *octave* mający za zadanie długotrwale wykorzystywać pełną moc obliczeniową procesora

```
for k=1:200000000
   for j=1:200000000
      for f=1:200000000
         n=k/(45678+k+j+f);
         m=n*k/(657+k);
      end
   end
end
```

-xf cpupw-0.2.0.tgz, następnie wejść do katalogu *cpupw-0.2.0/src* i wykonać polecenie *make* oraz *make install*. Nastąpi skompilowanie programu i instalacja pliku wykonywalnego programu w katalogu */usr/sbin*. Program do prawidłowego działania potrzebuje prawa odczytu i zapisu do pliku *msr* umieszczonego w katalogu */dev/cpu/n/*, gdzie *n* to numer procesora lub rdzenia. Jeśli jest więcej procesorów, to katalogów może być więcej niż jeden. Warunkiem prawidłowej współpracy programu *cpupw* z plikiem *msr* jest wkompilowana obsługa rejestrów MSR procesora na stałe do jądra lub w postaci modułu. MSR można włączyć w konfiguracji jądra w sekcji *Processor type and features* wybierając opcję */dev/cpu/*/msr - Model-specific register support*. Dla przypomnienia dodam, że konfigurację jądra można zmienić za pomocą polecenia *make xconfig* wywołanego w katalogu */usr/src/linux*, ze źródłami jądra. Polecam włączenie obsługi MSR jako modułu, gdyż zmniejszy to rozmiar jądra wynikowego. Po włączeniu MSR należy pamiętać o zbudowaniu nowego jajka i jego instalacji w menedżerze startu. Dla niewtajemniczonych lub tych co zapomnieli przypomnę, że można zrobić to za pomocą następującego ciągu poleceń: *make bzImage; make modules; make modules_install; make install*. W domyślnej konfiguracji Mandrivy 2008 zarówno w wersji desktopowej jak i laptopowej nie ma konieczności wprowadzania jakichkolwiek zmian w jądrze, gdyż konfiguracja domyślna jest wystarczająca. Gdy jest już uruchomione jądro z obsługą MSR można przejść do wyboru minimalnej wartości napięcia rdzenia procesora.

Wybór najniższej wartości napięcia rdzenia procesora

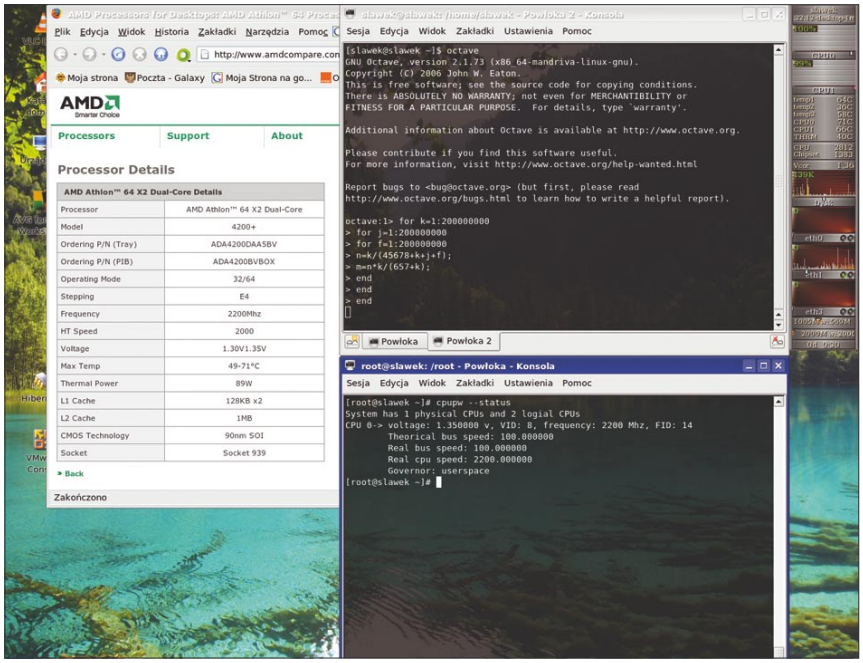
Należy załadować do pamięci moduł *msr* za pomocą polecenia *modprobe msr* z uprawnieniami administratora. Jeśli się tego nie zrobi program *cpupw* będzie zgłaszał błędy odczytu i zapisu pliku *msr* i jego działanie pozostanie bez skutku. Gdy moduł *msr* jest załadowany, za pomocą polecenia *cpupw --status* można zobaczyć aktualne ustawienia procesora, w tym aktualną wartość napięcia rdzenia, VID – identyfikator napięcia rdzenia, częstotliwość pracy procesora oraz FID – identyfikator częstotliwości i kilka innych informacji. Teraz można utworzyć swój pierwszy plik konfiguracyjny. Na początku najlepszy będzie plik wygenerowany automatycznie. Można go otrzymać po wydaniu polecenia *cpupw --autoconfig*. Wynik działania programu należy skopiować do pliku */etc/cpupw/config*. Przed wprowadzeniem nowych ustawień z pliku konfiguracyjnego, należy wyłączyć wszelkie usługi odpowiedzialne za oszczędzanie energii, tj. *cpufreq*, *cpudyn*, *powersaved* i inne zainstalowane w systemie oraz usunąć z pamięci moduł *acpi_cpufreq*. Warto również wyłączyć automatyczne uruchamianie tych usług przy starcie systemu lub zmienić konfigurację tak, by nie ingerowały one w energooszczędność procesora. Nowe ustawienia należy przetestować za pomocą polecenia *cpupw --config /etc/cpupw/config -f -s --smart*. Teraz, gdy działa oszczędzanie energii procesora w wersji domyślnej należy uruchomić jakiś program, który będzie intensywnie wykorzystywał moc obliczeniową wszystkich procesorów zainstalowanych w komputerze, ale nie będzie operował na ważnych dla nas danych. Można do tego celu użyć np. skryptu obliczeniowego (patrz Listing 1) w programie *octave*, dostępnym niemal w każdej dystrybucji w postaci pakietu instalacyjnego. Należy uruchomić tyle wersji programu *octave* wraz ze skryptem,

Listing 2. Skrypt *cpupw* z katalogu */etc/init.d/* uruchamiający program CPU Power przy każdym starcie systemu

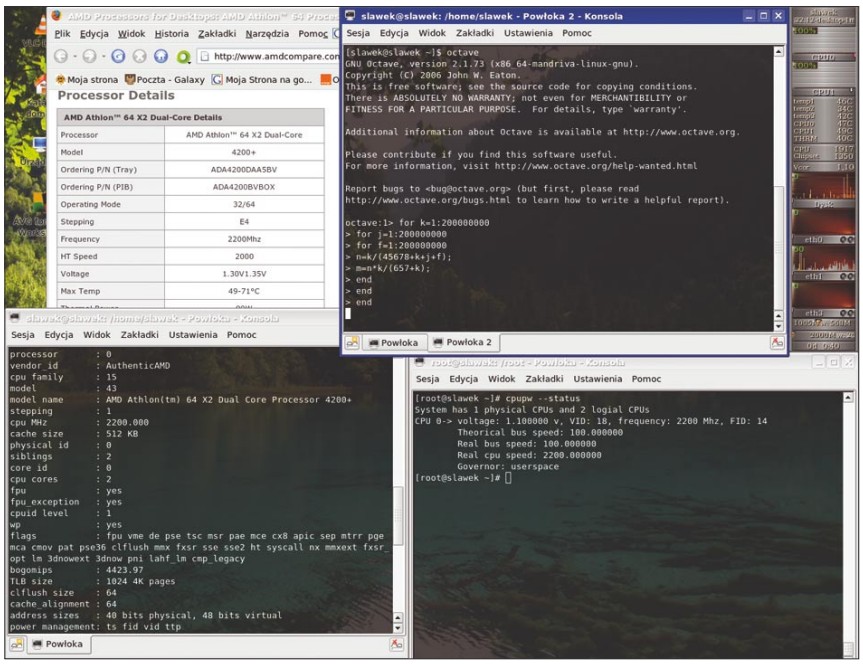
```
#!/bin/bash
#
### BEGIN INIT INFO
# Provides: cpupw
# Should-Start: acpi
# Should-Stop: acpi
# Default-Start: 2 3 4 5
# Short-Description: Set CPU frequency and voltages
settings
# Description: Set CPU frequency and voltages settings
### END INIT INFO
# Source function library.
. /etc/rc.d/init.d/functions
RETVAL=0
start() {
   gprintf "Setting CPU frequency settings: "
   touch /var/lock/subsys/cpupw
   test -f /etc/cpupw/config
   for cpu in /sys/devices/system/cpu/* ; do
       [ -f  $cpu/cpufreq/scaling_governor ] && echo
userspace > $cpu/cpufreq/scaling_governor
   done
   success
   gprintf "\nLoading 'msr' module: "
   modprobe msr
   success
   gprintf "\nStarting CPU Power: "
   /usr/sbin/cpupw -S -l -d --config /etc/cpupw/config
   success
   echo
}
real_stop() {
   gprintf "Resetting CPU frequency settings: "
   for cpu in /sys/devices/system/cpu/* ; do
       [ -f  $cpu/cpufreq/scaling_governor ] && echo
userspace > $cpu/cpufreq/scaling_governor
   done
   killall /usr/sbin/cpupw
   rm -f /var/lock/subsys/cpupw
   success
   echo
}
case "$1" in
   start)
      start
      ;;
   stop)
      ;;
   restart|reload)
      real_stop
      start
      ;;
   condrestart)
      if [ -f /var/lock/subsys/cpupw ]; then
         restart
      fi
      ;;
   status)
      ;;
```

ile jest rdzeni w komputerze lub innego programu, który całkowicie wykorzysta dostępna moc obliczeniową procesora. W momencie, gdy rozpoczną się obliczenia w oknie konsoli, w której jest uruchomiony *cpupw* będą ukazywać się informacje o zmianach w prędkości procesora. W osobnej konsoli można uruchomić sobie polecenie *cpupw --status*, aby zobaczyć aktualną prędkość procesora oraz przyporządkowane im napięcie rdzenia w danej chwili. Za każdym razem, gdy zaistnieje potrzeba sprawdzenia wyżej wspomnianych parametrów CPU (*ang. Central Processing Unit* – procesor) należy uruchomić powyższe polecenie. Jest to dosyć niewygodne. Lepszym rozwiązaniem będzie uruchomienie graficznego programu monitorującego, np. *gkrellm*, który pokaże aktualne informacje o procesorze w jednym miejscu, odświeżane w trybie rzeczywistym. *Gkrellm* do prawidłowej pracy wymaga instalacji pakietu *lm_sensors*, który do właściwego działania wymaga wykrycia wszystkich dostępnych czujników w systemie za pomocą polecenia `sensors-detect`, zawartego pakiecie `lm_sensors`. Wszystkie znalezione czujniki będą dostępne w *gkrellm* po wykryciu i uruchomieniu skryptu *lm_sensors* za pomocą polecenia `/etc/init.d/lm_sensors restart`.

Obserwacja zmian parametrów procesora pozwala zauważyć, że wraz ze wzrostem wykorzystania mocy obliczeniowej następuje zwiększenie prędkości procesora poprzez zwiększanie mnożnika oraz zwiększenie podawanego na niego napięcia (maleje VID) aż do maksymalnych wartości przewidzianych przez producenta. Mając już tę podstawową wiedzę o zachowaniu się procesora można pokusić się o próbę zmniejszenie napięcia rdzenia. Na początku zmiany należy wprowadzać stopniowo, zwiększając VID tylko o jeden od najmniejszego. Należy pamiętać, że wartość VID powinna znajdować się w przedziale przewidzianym przez producenta, tj. być mniejsza lub równa od VID odpowiadający najmniejszej częstotliwości i większa bądź równa wartości odpowiadającej największej dopuszczalnej częstotliwości procesora, gdyż w przeciwnym wypadku nie uda się ustawić niektórych wartości w procesorze, a dla za małego VID mieszczącego się poza przedziałem (duże napięcie) może nastąpić przegrzanie lub spalenie procesora. Nowe ustawienie należy zapisywać w pliku, osobno dla każdej prędkości (patrz ramka). Ustawienia zaczną obowiązywać dopiero po przerwaniu działania polecenia `cpupw --config /etc/cpupw/config -f -s -smart` np. za pomocą naciśnięcia klawiszy [*Ctrl+C*] i ponownym jego uruchomieniu. Daną wartość napięcia można uznać za dopuszczalną dopiero wtedy, gdy procesor będzie w stanie przepracować co najmniej 30 sekund bez zawieszenia systemu. Dla całkowitej pewności, że wybrane napięcie pozwala na stabilną pracę systemu, należy testować komputer przynajmniej 30 minut przy pełnym obciążeniu dla danych ustawień. Jeśli



Rysunek 1. Maksymalne obciążenie procesora AMD Athlon 64 X2 4200+ w domyślnej konfiguracji, przy napięciu 1,35 V przy częstotliwości 2200 MHz



Rysunek 2. Maksymalne obciążenie procesora AMD Athlon 64 4200+ przy obniżonym napięciu rdzenia – Vcor = 1,1 V dla częstotliwości 2200 MHz

Jak obliczyć VID?

Wartość napięcia VID procesora można obliczyć za pomocą wzoru różniącego się w zależności od producenta układu. Dla procesorów firmy AMD (Athlon 64, Turion, Opetron) wzór ma postać: *VID = (1550 – napięcie_w_mV)/25*, a dla procesorów firmy Intel (Pentium-M, Celeron-M oraz wszystkich wykonanych w technologii Centrino): *VID = (napięcie_w_mV – 700)/16*. Z identyfikatora VID można po przekształceniu wyżej podanych wzorów obliczyć wartość napięcia w miliwoltach, dla AMD: *napięcie_w_mV = 1550 – (VID * 25)*, a dla Intela: *napięcie_w_mV = (VID * 16) + 700*.

po tym czasie system się nie zawiesi to będzie to oznaczać, że nowe ustawienia są prawidłowe. Nowe wartości napięć pozwolą zaoszczędzić energię i zmniejszyć temperaturę procesora bez zmniejszania jego możliwości obliczeniowych. Aby ustawienia były dostępne na stałe po każdym uruchomieniu systemu, należy z uprawnieniami administratora w katalogu */etc/init.d/* utworzyć plik *cpupw* o zwartości jak w Listingu 2, nadać mu prawo odczytu, zapisu i wykonywania dla użytkownika oraz włączyć uruchamianie przy każdym starcie systemu, np. za pomocą *Centrum Sterowania* Mandriva Linux.

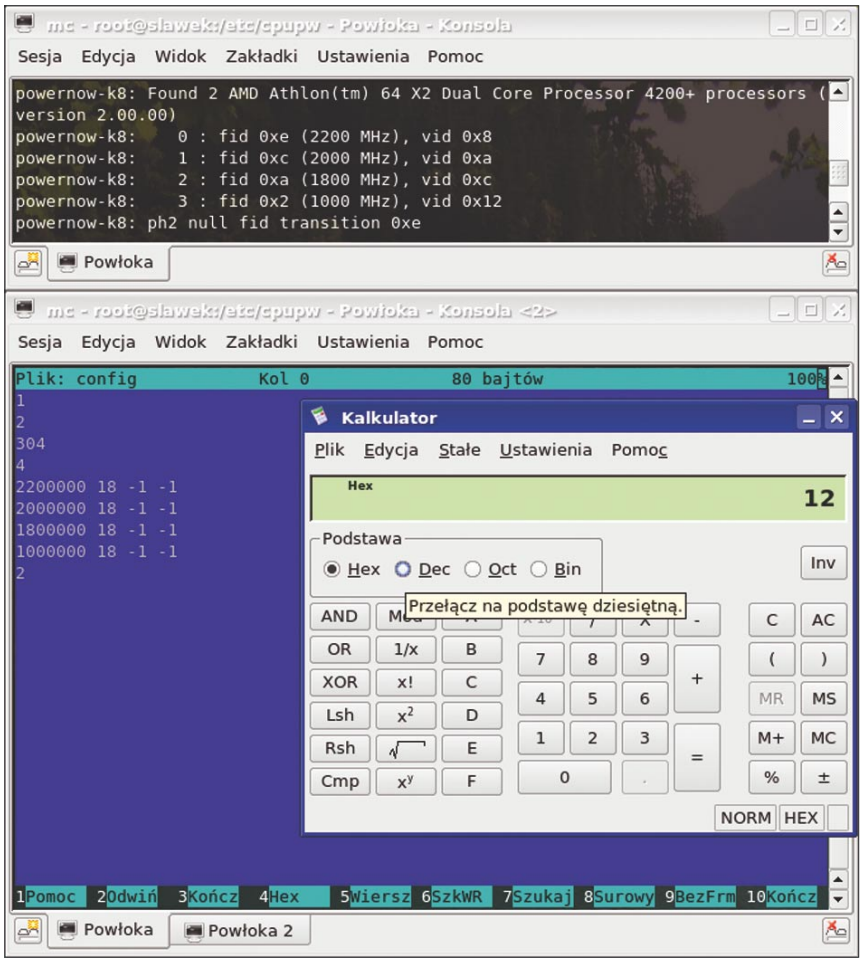
Wyniki badań testowych

Zaprezentowany powyżej sposób oszczędzania energii został przetestowany w procesorze AMD Athlon 64 X2 4200+ z rdzeniem Manchester w technologii 90 nm na podstawce 939. Do badań posłużył komputer o następującej konfiguracji sprzętowej:

- procesor: AMD Athlon 64 X2 Dual Core 4200+ ADA4200DAA5BV (dokładne dane – patrz Rysunek 1),
- płyta główna: Asus A8N-E, podstawka pod procesor: 939, Nvidia NForce 4 Ultra,
- pamięć: Apacer 2 * 512 MB, DDR, PC400, CL2,5,
- dysk twardy: WD Caviar 500 GB WD5000YS-01MPB1 09.02E09, 7200 rpm, SATA II,
- napęd optyczny: LiteOn 1693S,
- karta graficzna: Sapphire ATI X300SE z chłodzeniem pasywnym,
- obudowa: Codegen 350 W,
- klawiatura: Genius KB19E,
- myszka: A4Tech SWOP-35,
- system operacyjny: Mandriva 2008 Free DVD x86_64 z jądrem 2.6.22.12.

Tabela 1. Porównanie poboru mocy całego komputera z procesorem AMD Athlon 64 X2 4200+ dla domyślnego oszczędzania energii oferowanego przez program Cpufreq i dla obniżonego napięcia zasilania realizowanego za pomocą programu CPU Power

Częstotliwość procesora [MHz]	Cpufreq		CPU Power	
	Moc komputera [VA]	Napięcie zasilania rdzenia procesora [V]	Moc komputera [VA]	Napięcie zasilania rdzenia procesora [V]
2200	142,6	1,35	98,9	1,1
2000	124,2	1,3	96,6	
1800	110,4	1,25	94,3	
1000	82,8	1,1	82,8	



Rysunek 3. Domyślne wartości VID i FID dla dostępnego zakresu częstotliwości w procesorze AMD Athlon 64 X2 4200+ (u góry)

Skąd wziąć domyślne wartości napięć procesora?

Są dwa sposoby znalezienia domyślnych wartości napięć procesora. W przypadku procesorów AMD 64 wystarczy, załadować moduł *powernow_k8* poleceniem *modprobe powernow_k8*, a następnie wydać polecenie *dmesg|grep powernow-k8*, aby uzyskać informację o wszystkich dostępnych prędkościach oraz przyporządkowanych im kodach FID i VID zapisanych w kodzie szesnastkowym (hex). Otrzymane kody należy przeliczyć, np. za pomocą kalkulatora *kcalc* do postaci dziesiętnej (dec). W przypadku procesorów Intel działających pod kontrolą najnowszych jąder po załadowaniu modułu odpowiedzialnego za oszczędzanie energii nie ukaże się w logach systemowych informacja o dostępnych trybach. Informacji można poszukać w Internecie, albo sprawdzić to ręcznie. Należy upewnić się, że nie działa żadna kopia programu *cpupw* w systemie i uruchomić domyślne oszczędzanie energii, np. za pomocą skryptu */etc/init.d/cpufreq start*, a następnie wykonać polecenie `cat /sys/devices/system/cpu/cpuN/cpufreq/scaling_available_frequencies` za pomocą którego nastąpi wypisanie dostępnych prędkości procesora – *N* to numer fizycznego procesora liczony od zera. Potem trzeba obciążyć procesor obliczaniami, np. za pomocą skryptów w programie *octave* (po jednym skrypcie na procesor) i zmienić kolejno maksymalną dopuszczalną częstotliwość procesora na każdą z wypisanych przez poprzednie polecenie komendą `echo „prędkość_procesora_w_kHz”>/sys/devices/system/cpu/cpuN/cpufreq/scaling_max_freq` i za pomocą programu *cpupw –status* sprawdzić jakie napięcie (kod VID) odpowiadają danej częstotliwości.

Zużycie energii elektrycznej zostało zmierzone cyfrowym miernikiem elektronicznym DT9208A. Na początku zmierzono napięcie panujące w sieci (U=230V), a następnie wartość pobieranego prądu (I) przez komputer w wymienionej konfiguracji. Otrzymane wartości prądu zostały przeliczone na moc całkowitą ze wzoru P=U*I uwzględniającego pobór mocy czynnej (energii w watach) i biernej (którą można w przybliżeniu określić jako energię strat). Otrzymane wyniki okazały się zaskakujące. Komputer po starcie, przy włączonym oszczędzaniu energii i przy braku jakiegokolwiek wykorzystania mocy obliczeniowej zużywał 66,7 VA (≈W). Po uruchomieniu dwóch kopii programu *octave* (po jednym na każdy z dwóch rdzeni) i włączeniu skryptu obliczeniowego wykorzystującego całkowicie moc obliczeniową dwóch rdzeni, przy domyślnym napięciu rdzenia procesora, ustawionym za przez program *cpufreq* (domyślna polityka oszczędzania energii) wynoszącym 1,35 V zużywał 142,6 VA. (patrz Rysunek 1)

Po lewej stronie widać pełną specyfikację procesora, na dole ustawienia procesora odczytane przez program *cpupw –status*, na środku uruchomiony skrypt w programie *octave* wykorzystujący całkowicie moc obliczeniową procesora, a po prawej program *gkrellm* pokazujący temperaturę pierwszego rdzenia – CPU0 (równa maksymalnej dopuszczalnej), drugiego rdzenia – CPU1 (bliska dopuszczalnej), prędkość obrotową wentylatora na procesorze – CPU (prawie maksymalną osiągalną przez wiatraczek) i napięcie rdzenia – Vcor.

Dla takiego samego obciążenia obliczeniowego, ale napięcia rdzenia wynoszącego już tylko 1,1 V, najniższego stabilnego (*undervolting*) ustawionego przez CPU Power udało się zmniejszyć moc całego komputera o 30,7%, czyli do 98,9 VA.(Rysunek 2)

Po prawej stronie w programie *gkrellm* widać o ponad 20ºC niższą temperaturę pierwszego i drugiego rdzenia – CPU0 i CPU1, blisko o 1000 obrotów na minutę mniejszą prędkość obrotową wentylatora na procesorze – CPU – niż dla napięcia rdzenia 1,35 V (Rysunek 1).

Równocześnie wraz ze spadkiem poboru mocy nastąpił spadek ilości wydzielanej energii cieplnej przez procesor, co objawiło się spadkiem temperatury rdzenia z 71ºC do 49ºC dla tego samego obciążenia obliczeniowego i równoczesnym spadkiem prędkości obrotowej wentylatora na procesorze oraz redukcją generowanego hałasu. Pozwoliło to zwiększyć odstęp temperatury pomiędzy maksymalnym obciążeniem obliczeniowym, a maksymalną dopuszczalną wartością temperatury określoną przez producenta (71ºC).

Czytelnikowi wyda się zapewne interesujące dlaczego przy domyślnych ustawieniach temperatura doszła do wartości krytycznej podanej przez producenta. Było to możliwe, ponieważ na procesorze dwurdzeniowym o mocy termicznej (89 W) zainstalowano radiator do jednordzeniowego procesora o mocy termicznej (60 W). Po zastosowaniu *undervoltingu* radiator od mniej energochłonnego procesora okazał się wystarczający. Dla potrzeb niniejszego przykładu zastosowano mniej wydajne chłodzenie po to by pokazać, jak dużo energii marnuje się na podgrzanie rdzenia procesora. Nie zaleca się jednak stosowania słabszego chłodzenia niż zalecane przez producenta dla danego typu procesora. Można jedynie pokusić się o wykorzystanie wydajniejszego układu chłodzenia.

Napięcie procesora można obniżyć również za pomocą innych programów, np. za pomocą Linux-PHC (*http://linux-phc.sourceforge.net*). Niestety część z nich, jak również Linux-PHC wymaga łatania jądra systemowego i częstokroć ma zastosowanie tylko dla procesorów jednego producenta. Mimo tych trudności warto próbować zwiększać energooszczędność swoich komputerów, albo z powodów czysto materialnych, jak niższe rachunki za energię elektryczną, czy dłuższy czas pracy na baterii lub bardziej wzniosłych, jak ochrona środowiska przed zanieczyszczeniami generowanymi przez elektrownie. Gdy na komputerach z zainstalowanymi dwoma systemami operacyjnymi zaistnieje konieczność skorzystania z systemu Windows, to w celu zachowania oszczędzania energii wykorzystującego *undervolting* należy zainstalować np. program Notebook Hardware Control (*http://www.pbus-167.com*) dla Windows i skonfigurować go zgodnie z wartościami napięcia otrzymanymi doświadczalnie w systemie Linux. Pozwoli to cieszyć się obniżonym zużyciem energii również w systemie Windows.

Redakcja i autor nie ponoszą żadnej odpowiedzialności za ewentualne uszkodzenia wynikłe ze zmiany napięcia zasilania procesora. Nie zaleca się również prowadzenia eksperymentów osobom, które nie mają w tym żadnego doświadczenia.

Budowa pliku konfiguracyjnego programu CPU Power

Na podstawie poniższego przykładu zostanie omówiona budowa pliku konfiguracyjnego.

```
1
2
304
4
2200000 18 -1 -1
2000000 18 -1 -1
1800000 18 -1 -1
1000000 18 -1 -1
2
```

W linii pierwszej znajduje się liczba określająca ilość fizycznych procesorów, a w drugiej ilość jednostek logicznych w tym rdzeni HT w procesorach Intel. W trzeciej linii znajduje się etykieta sterownika jądra dla danego procesora, a w czwartej ilość dostępnych częstotliwości pracy procesora. W następnych liniach znajdują się wszystkie dostępne częstotliwości w MHzach od największej do najmniejszej, w ilości podanej w linii czwartej. Każda linia odpowiadająca prędkości w pierwszej kolumnie zawiera prędkość w MHzach, w drugiej kod napięcia rdzenia VID w kodzie dziesiętnym, w trzeciej górny, a w czwartej dolny próg wykorzystania mocy obliczeniowej procesora. W ostatniej linii pliku konfiguracyjnego znajduje się liczba określająca ilość logicznych procesorów w fizycznej jednostce CPU.

Wartość *-1* w danej kolumnie oznacza, że zostanie zastosowane ustawienie domyślne dla danej wartości, czyli najbezpieczniejsze z punktu widzenia stabilności działania procesora.

Samodzielne utworzenie pliku konfiguracyjnego może być dosyć kłopotliwe. Zaleca się automatycznie wygenerować plik za pomocą polecenia *cpupw –autoconfig*, zapisać go i dopiero wtedy przystąpić do wprowadzania w nim nowych niestandardowych wartości napięcia rdzenia procesora.

O autorze

Autor przez 3 lata administrował serwerem, natomiast na co dzień używa dystrybucji Mandriva.
Bardzo lubi eksperymentować z Linuksem.
Kontakt z autorem: *diodak@gmail.com*