

O projeto LessWatts ajuda o Linux a economizar energia

# Economizando velas

Os sistemas Linux já consomem menos energia hoje do que há um ano. No entanto, o projeto LessWatts mostra que ainda há um longo caminho à frente.

por Sulamita Garcia

**E**conomia de energia tem se tornado um novo foco na área de tecnologia. Talvez para você o consumo mensal de energia do seu computador não faça tanta diferença, mas para um data center com centenas ou milhares de computadores, além do custo direto da energia gasta pelos computadores, existe a energia gasta para resfriá-los, dissipando o calor que eles produzem. Ou, para habitantes de áreas distantes, significa poder usar seu computador por mais tempo, pois eles apenas têm energia para carregar a bateria três horas por dia.

O projeto *LessWatts* foi criado para reunir e fomentar iniciativas para melhoria do gerenciamento de energia no Linux. Todos os que

utilizam notebooks podem confirmar que ainda existe um longo caminho a ser traçado nesse sentido. O website do projeto contém ferramentas, documentação, dicas e guias de desenvolvimento de software utilizando técnicas que evitam o gasto desnecessário de energia. Lançado há cerca de um ano, no início, o projeto consistia da ferramenta *PowerTop* e de alguns documentos. É surpreendente o crescimento do projeto, que hoje conta com 12 subprojetos e extensa documentação técnica abordando diferentes elementos envolvidos no consumo de energia no Linux. Porém, logo ficou claro que, embora o consumo de energia do kernel tenha melhorado muito, as aplicações em torno do Linux tinham muito o que

melhorar para que o sistema fosse eficiente nesse quesito.

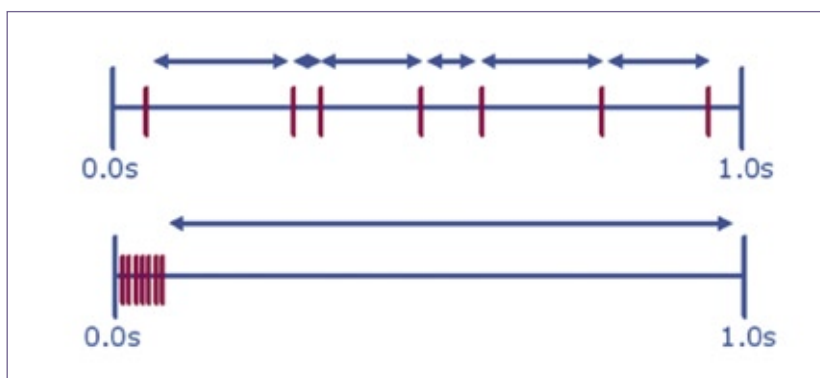
Assim, vários projetos receberam e continuam recebendo melhorias, além do trabalho de documentar e criar nos desenvolvedores uma nova mentalidade para desenvolvimento de software.

Vamos examinar alguns desses projetos, bem como algumas dicas para se ter em mente na hora de desenvolver software.

## Otimização economiza energia

Recentemente, a Xandros anunciou a adoção da plataforma Moblin para seus netbooks e, com isto, aumentou em cerca de 25% a duração da bateria dos dispositivos. Isso representa uma hora a mais, apenas com modificações de software disponíveis como código aberto. Vejamos algumas formas de conseguir isso:

- ▶ Evitar *polling* – a maioria das aplicações em uso hoje em dia foi implementada com base em um kernel com ciclos periódicos. Essas aplicações supõem que o kernel acordará cada CPU múltiplas vezes por segundo. Como resultado, o *polling* se tornou uma solução fácil e simples para muitas aplicações. Cada vez que uma aplicação



**Figura 1** Ao agruparmos timers, conseguimos fazer com que o processador seja acionado durante um tempo menor e permaneça por mais tempo no estado ocioso.

## Quadro 1: Abuso do polling

Certas aplicações realizam polling intensamente durante sua operação. Algumas dessas ações e os responsáveis por elas:

- ▶ checar uma vez por segundo se o mouse se moveu (*gnome-screensaver*)
- ▶ checar dez vezes por segundo se o volume do som foi modificado (*mixer applet*)
- ▶ checar uma vez por segundo se é hora de mostrar o próximo minuto no relógio (*clock applet*)
- ▶ checar 30 vezes por segundo (*gamin*) ou dez vezes por segundo (*Evolution*) se existe algum dado na fila interna
- ▶ checar dez vezes por segundo se existe um leitor de smartcard inserido na USB (*gdm-daemon*)
- ▶ checar mil vezes por segundo se existem dados em um pipe (*gksu*)

Polling é um recurso válido para várias aplicações, mas tenha em mente que seu uso indiscriminado pode gerar consumo desnecessário de energia.

faz uma requisição à CPU por meio do kernel, a CPU acorda do estado de inatividade, consumindo energia. Um processador atual consome 34 watts por segundo quando funcionando em carga total, e apenas 1 quando ocioso. O **quadro 1** lista exemplos de aplicações reais que abusam do polling.

- ▶ Adiantar a entrada no modo ocioso utilizando a velocidade máxima do processador: processadores tendem a economizar mais energia quando ociosos; então, geralmente é melhor executar uma operação o mais rápido possível para poder permanecer inativo por mais tempo. Vejamos um exemplo: no processador mencionado antes, decodificar um segundo de um arquivo MP3 leva meio segundo usando metade da velocidade máxima e 0,25 segundo a toda velocidade. Então, o consumo de energia para essa tarefa é:

Metade da velocidade:  $0,5s \times 24W + 0,5 \times 1W = 12,5$  Joules

Velocidade Máxima:  $0,25s \times 34W + 0,75 \times 1W = 9,25$

- ▶ Desligar dispositivos que não estejam sendo usados: esses dispositivos, além de consumirem energia desnecessariamente, impedem que o sistema entre em modo de economia de energia.
- ▶ Agrupar *timers*: muitos programas usam timers, então agrupá-los ajuda a reduzir as chamadas que tiram o processador do modo ocioso, como mostra a **figura 1**.
- ▶ Atentar ao uso de linguagens de alto nível: essas linguagens são ferramentas convenientes para a obtenção de resultados rapidamente, e freqüentemente têm funcionalidades que permitem executar tarefas complexas com o mínimo de esforço. Entretanto, esteja ciente de que algumas dessas estruturas são difíceis de implementar e algumas vezes o ambiente de execução associado à linguagem empregam polling em alta freqüência.

Ao usar linguagens tais como *Java*, *Visual C#*, *Python* e *Ruby*, cheque o resultado final e evite usar algumas das estruturas de *threading* mais primitivas. E quando existir a possibilidade de escolha do ambiente de execução, avalie diferentes alternativas e versões.

## PowerTop

Uma das principais ferramentas do projeto LessWatts é o PowerTop, uma aplicação simples cujo objetivo é medir os vilões de consumo de energia no sistema. Além de mostrar quais aplicações estão consumindo mais recursos, ele mostra dicas para melhorar ainda mais a economia de energia. Se você quiser, basta acionar as teclas que são listadas junto com as dicas para que o PowerTop faça as mudanças de forma autônoma.

Assim que foi lançado, o PowerTop logo começou a mostrar os aplicativos que costumavam estar no topo da lista de esbanjadores de energia. O *Pidgin*, antigo *Gaim*, foi um dos primeiros a receberem melhorias e, na versão 2.0.1, já deixava o “Muro da Vergonha”. O *Firefox* ainda consta na lista, mas também vem recebendo melhorias para deixar essa condição.

O projeto LessWatts é uma comunidade que serve de ponto de troca de conhecimento entre todos os interessados em melhorar o desempenho dos sistemas de código aberto no consumo de energia. Nele, você pode encontrar diversos documentos, tutoriais, ferramentas de avaliação do seu sistema e muito mais. Além disso, pode participar e contribuir diretamente para essas melhorias. Não deixe de visitá-lo. ■

### Mais informações

[1] LessWatts: <http://www.lesswatts.org/>