

Administre seu celular com a dupla Gammu e Wammu

Auxílio à lista

◉ Gammu lê e modifica livros de endereços, agendas e até as configurações de aparelhos celulares – tudo isso na linha de comando. Se você não gosta da telinha preta, seu “primo” Wammu será um bom companheiro para alegrar seu desktop enquanto fuça no seu telefone.

POR CARSTEN SCHNOBER

Nessa era cada vez mais móvel, o telefone celular tornou-se a central pessoal de comunicações do indivíduo. Não é mais um simples telefone: ele lembra a seu mestre que ele tem que pagar o dízimo ou a hora da reunião com o chefe do chefe, além de armazenar telefones e endereços. De tempos em tempos, nada melhor do que sincronizar as informações do telefone com as do computador de mesa.

O Gammu [1], um programinha em modo texto, pode acessar praticamente todos os dados armazenados em uma infinidade de aparelhos celulares – e não se importa se você liga o telefone ao micro usando um cabo USB, uma conexão por infravermelho ou Bluetooth. Como a maioria dos donos de celulares prefere evitar a linha de comando ao máximo, o Wammu [2] é o companheiro ideal para o Gammu, oferecendo uma interface gráfica para intermediar o “trabalho sujo” que é usar o Gammu.

Instalação

O SUSE Linux inclui tanto o Gammu como o Wammu no CD de instalação. Os usuários dessa distribuição alemã só precisam chamar o YaST para instalar as duas jóias. Os usuários de outras distribuições precisam consultar suas fontes usuais de pacotes para obter pacotes RPM, Debian ou mesmo o código fonte.

Baixe o pacote RPM e instale-o com o comando

```
rpm -Uvh gammu-1.01.11-1.te.i386.rpm
```

Se for um pacote *.deb*, o comando apropriado é

```
dpkg -i libgammu0_1.01.11-0_i386.deb gammu_1.01.11-0_i386.deb
```

Você precisará, para o pacote RPM, da biblioteca *bluez-libs*. Se seu sistema for um Debian, a biblioteca necessária é a *libbluetooth1*. Ambas instalam um subsistema Bluetooth apropriado para a distribuição em uso.

O *front-end* Wammu depende do pacote *python-gammu* [3]. Baixe o arquivo e rode os seguintes comandos:

```
tar -xjf python-gammu-0.8.tar.bz2 # Para descompactar.
cd python-gammu-0.8 # Vai para o diretório recém-criado.
python setup.py build # Prepara o Gammu...
python setup.py install # ... e o instala em seu sistema.
```

O procedimento para instalar o Wammu propriamente dito é bastante parecido.

Wammu

Com tudo instalado, podemos começar. O comando `wammu` inicia a interface gráfica. Na primeira vez em que é chamado, o `wammu` não encontra as configurações pessoais do usuário em seu diretório `/home` e, portanto, mostra uma caixa de diálogo para esse ajuste. Se você já possuir um arquivo `~/ .gammurc`, o Wammu pega os valores padrão de lá. Se não possuir, o programa pergunta se o usuário deseja que sejam detectados automaticamente, verificando as interfaces mais comuns à cata de dispositivos conectados.

Se o Wammu não for capaz de detectar seu fone automaticamente, será preciso informar manualmente os dados do aparelho no menu `Wammu | Settings` (figura 1). Digite o nome da interface no campo `Device`: costuma ser `/dev/ircomm0` para interfaces infravermelhas, `/dev/hci0` para Bluetooth e `/dev/ttyS1` se for usado o cabo USB que veio junto com o telefone. Dependendo de como sua distribuição lida com as configurações, esses nomes todos podem ser ligeiramente diferentes. Consulte o [quadro 1](#) para mais informações sobre como configurar uma interface desse tipo. Consulte também o artigo *Surfando no ônibus*, na página 24 desta edição, mais especificamente o [quadro Conectando seu celular via Bluetooth](#).

Defina um protocolo na seção `Connection`. Em [4] há uma lista de telefones celulares e tipos de comunicação com os quais o Gammu funciona. Depois, escolha `Phone | Connect` para que o Wammu se conecte ao celular. A barra de `status` no rodapé da janela mostra o estado da bateria, a hora interna e o nível do sinal que o telefone está recebendo.

O menu `Retrieve` serve para baixar os dados do telefone para o computador. Nem todos os aparelhos respondem a todas as funções desse menu – consulte novamente o banco de dados em [4] para mais detalhes. O menu `Info` mostra os

detalhes técnicos do celular, como número de série, modelo e versão do firmware.

O menu `Contacts` dá acesso ao livrinho de endereços e telefones, que pode ser organizado em dois grupos: itens no cartão SIM ou itens na memória do telefone. De forma similar, o menu `Calls` recupera o histórico de chamadas, `Messages` lê as mensagens SMS, `Todos` lê a lista de tarefas a cumprir (*to do list*) e `Calendar` lê sua agenda de compromissos.

Ler é fácil, já gravar... Nem todos os recursos de envio de dados ao telefone estão implementados para cada tipo de aparelho na versão 0.8 do Wammu. Em nosso laboratório, fomos incapazes de criar novos contatos ou compromissos. Novas tarefas também provocaram erros, mas mesmo assim os dados foram guardados na memória do telefone. Uma história parecida ocorreu com o editor de SMS mostrado na figura 2. Apesar do erro, o torpedo SMS foi realmente entregue ao destinatário.

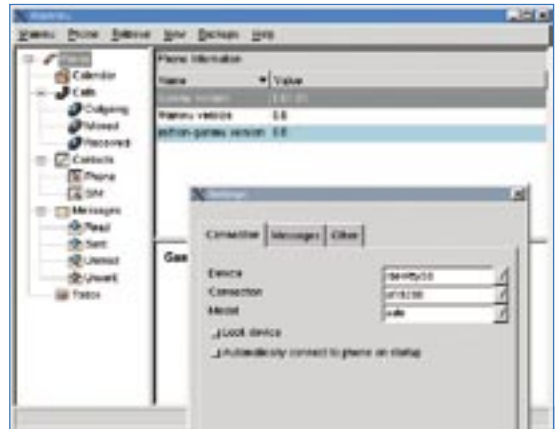


Figura 1: A interface do Wammu permite que os usuários acessem os telefones, endereços e compromissos cadastrados no celular.

Cópia de segurança

Um backup com todas os dados do telefone – normalmente importantes – evita que a perda ou roubo do aparelho seja catastrófica. O menu `Backup` o ajuda a gravar uma cópia de segurança de todos os dados do telefone, ou mesmo de uma parte deles. O menu `Backup | Save` abre um gerenciador de arquivos no qual podemos especificar o formato de exportação dos dados. ➔

Quadro 1: Telefones celulares com interface infravermelha

Muitos laptops possuem uma interface infravermelha, e também há adaptadores infravermelhos que se pode ligar a uma porta USB. Ao contrário dos cabos, que são específicos para um determinado telefone, um adaptador infravermelho tem a vantagem de poder ser usado com aparelhos de diferentes procedências. Já o caso do Bluetooth é diferente: há muitos computadores e telefones que não o reconhecem ainda, portanto não podemos dizer que seja uma alternativa popular.

Comece instalando o pacote `irda` – algumas distribuições o chamam de `irda-utils`. Se seu adaptador infravermelho for USB, não esqueça de reiniciar o subsistema infravermelho com o comando `/etc/init.d/irda restart`. Você precisa ser `root` para fazer isso.

Muitas distribuições associam por padrão a interface infravermelha ao dispositivo `/dev/ttyS1`. Se você possuir um adaptador FIR (“Fast Infrared”, infravermelho de alta velocidade) que seja reconhecido pelo Linux, é preciso alterar o arquivo de configuração. No lugar de `/dev/ttyS1`, deve-se indicar o tipo da interface, que se parece com algo como `sti4200` – esse é só um exemplo, verifique o nome real em seu sistema. O arquivo que lida com essas configurações no SUSE é chamado `/etc/sys-config/irda`, mas nomes como `/etc/irda` ou `/etc/default/irda-utils` (este último, do Debian e derivados) são bastante comuns.

Agora que já sabemos qual é a porta de nosso dispositivo infravermelho, a próxima etapa é informar ao Gammu (e, por tabela, ao Wammu) que porta é essa. Na configuração do Gammu e do Wammu, especifique o item `Port` com a porta que acabamos de descobrir, seja ela `/dev/ttyS1` ou o nome da interface FIR. Adicione uma linha apropriada na seção `Connection` – tipicamente `irdaphonet`, embora também sejam comuns `infrared`, `fbusirda` e `irdaat`.

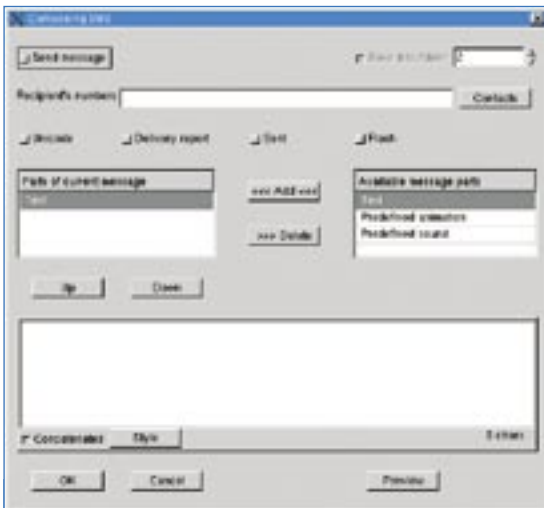


Figura 2: Use o editor de mensagens SMS para escrever torpedos sem o pesadelo de ter que digitá-los no “teclado” minúsculo do telefone.

A opção *Gammu backup [all data]* armazena todos os dados no formato padrão do Gammu. Subconjuntos dos dados podem ser gravados nos populares formatos *LDIF* e *VCF*. A maioria dos aplicativos reconhece pelo menos um deles. Para dispositivos Nokia, há até a opção de usar o formato proprietário da empresa. O Wammu também processa dados de agenda para que outros programas possam usá-los, como o formato *iCal*, da Apple, ou o padrão aberto *vCal*.

Gammu

Os desenvolvedores do Wammu (ainda) não implementaram todos os recursos que os telefones celulares possuem, mas a ferramenta Gammu, que funciona na linha de comando, preenche os buracos na parede com um pouco de rejunte. O Gammu permite que os usuários leiam o sistema de arquivos do telefone e controlem os recursos especiais – normalmente rádios, reproduzidores de MP3 e câmeras. É possível inclusive rodar testes para verificar a campainha vibratória e a iluminação do teclado e do *display*.

Para ajustar o comportamento do Gammu basta editar o arquivo `.gammurc`, presente em sua pasta pessoal (diretório

`/home`). Se desejar configurar as opções para todos os usuários, é preciso mexer no arquivo `/etc/gammurc` – ao menos, se você tiver instalado o programa pelo pacote RPM. Se seu sistema for o Debian, copie o arquivo comprimido que se encontra em `/usr/share/doc/gammu/examples/config/gammurc.gz` para seu diretório pessoal. Descompacte-o com o comando `gunzip gammurc.gz` e renomeie-o para `.gammurc` com este comando: `mv gammurc .gammurc`.

Para usar seu telefone, você pode se tornar *root* e editar o arquivo de configuração global ou permanecer como usuário normal e editar sua configuração pessoal. Os itens `port` e `connection` na seção `[gammu]` do arquivo são as de interesse imediato. Como era de se esperar, uma cerquilha (`#`) faz o Gammu ignorar aquela linha. A configuração é bem parecida com a do Wammu, que vimos antes. Você pode novamente consultar [4] ou usar os valores mais comuns: `/dev/ttyS1` para `port`, `/dev/rfcomm0` para Bluetooth ou `/dev/ttyACM0` para conectar pelo cabo de dados. Também copie a seção `connection` da configuração do Wammu.

O comando `gammu -help` mostra uma lista enorme de opções para uso do Gammu, algumas das quais específicas para um dado aparelho de celular. Para refinar a busca por socorro, informe o tópico mais relevante, algo como `gammu --help filesystem`, por exemplo. O parâmetro `identify` informa se o Gammu consegue (ou não) falar com seu telefone:

```
$ gammu --identify
Fabricante : Nokia
Modelo    : 6610 (NHL-4U)
Firmware  : 4.74 C (14-04-03)
Hardware  : 0640
```

O Gammu possui todos os recursos do Wammu – e muitos mais na manga. O comando `gammu --getallcalendar` lê sua agenda e a mostra como texto puro na tela. O resultado mostrado depende muito dos recursos do seu aparelho. Por exemplo, alguns telefones mostrarão todo o conteúdo do livrinho de endereços quando emitimos o comando `gammu --getallcategory PHONEBOOK`. Já se o comando for `gammu --getcategory PHONEBOOK 1`, apenas o primeiro item do livrinho é mostrado.

Outros celulares esperam que se especifique a opção `--getmemory` seguido de um tipo de memória, onde `ME` é o livro de telefones interno do aparelho e `SM` é o cartão SIM. Pode-se ainda identificar um item, basta informar seu número ou usar uma faixa numérica. Por exemplo, o comando a seguir mostraria os itens de 1 a 100 do livro de endereços na memória interna do aparelho:

```
gammu --getmemory ME 1 100
```

O comando `gammu --getmemory DC 1 - 10` usa uma técnica semelhante para acessar os números chamados que foram armazenados em posições de memória – nesse caso, os números de 1 a 10. `DC` vem de *Dialed Calls*. A opção `MC` (*Missed Calls*) mostra as chamadas perdidas e `RC` (*Received Calls*) as chamadas recebidas.

Na mesma linha, vários truques são possíveis. `gammu --getmemory ON 1` mostra o próprio número do telefone (*Own Number*). Já `VM` lista as mensagens em sua caixa de entrada (*View Mailbox*). Para todas essas funções funcionarem será preciso, obviamente, que o telefone possua o recurso que você está tentando acessar.

Para pesquisar o sistema de arquivos do seu celular, digite `gammu --getfilesystem`. Uma lista com todos os arquivos guardados no telefone, separados por pasta,

será mostrada. É possível transferir um arquivo do seu telefone para o disco rígido, basta especificar o número:

```
gammu --getfiles 63
```

Até os fabricantes possuem documentação específica na ajuda online. Os comandos `gammu --help nokia` e `gammu --help siemens`, por exemplo, mostrarão mais detalhes sobre esses dois fornecedores. Novamente, nem todos os telefones reconhecem todos os comandos. Por exemplo, tentamos ler o código de segurança de alguns telefones Nokia: em muitos deles, em vez do código correto (que já sabíamos de antemão) recebemos `12345`. De qualquer forma, o comando para isso é `gammu -nokiasecuritycode` – você pode tentar aí em seu aparelho.

Se você já tiver aderido à mania dos *ringtones* (toques de telefone especiais), pode transferir seu toque preferido do telefone para o computador e vice-versa. Use o comando `gammu --help ringtone` para saber os parâmetros apropriados. O comando `gammu -getringtoneslist` mostra uma lista completa dos toques que já estão no telefone. Já o comando `gammu --getringtone 51 meu.toque.preferido` transfere o *ringtone* número 51 para seu disco rígido, num arquivo chamado `meu.toque.preferido`. O formato depende do seu aparelho.

Monitorando as chamadas recebidas

O Gammu permite que você monitore as chamadas que porventura receba enquanto estiver conectado ao telefone. O mesmo vale para torpedos SMS. O comando `gammu -monitor` mostra uma lista de eventos a intervalos regulares, semelhante à da [listagem 1](#).

Caso cheguem mensagens SMS ou chamadas, o modo monitor mostra os números de quem chamou. Você poderia, por exemplo, escrever um script em Bash para ler os dados, convertê-los em

Listagem 1: O Gammu em modo de monitoramento

```
gammu --monitor
Press Ctrl+C to break...
Entering monitor mode...

Enabling info about incoming SMS : Enabling info about incoming SMS : %s

Enabling info about incoming CB : Enabling info about incoming CB : %s\n
Enabling info about calls : Enabling info about calls : %s\n
Enabling info about USSD : Enabling info about USSD : %s\n\n

SIM phonebook : Used 017, Free 233
Dialled numbers : Used 020, Free -08
Received numbers : Used 010, Free 002
Missed numbers : Used 266, Free -254
Own numbers : Used 000, Free 005
Phone phonebook : Used 109, Free 191
ToDos : Used 2
Calendar : Used 47
Battery status : 100 percent
Network level : 100 percent
SIM SMS Status : 8 used, 0 unread, 40 locations
Phone SMS status : 15 used, 0 unread, 150 locations, 20 templates
Network reception: Home network
Network : 262 07 (Network : %s (%s", Germany), LAC C75F, CID 7563
Name in phone : "o2 - de"
```

um *log* e avaliar os resultados com um pouquinho de lógica de programação. Os desenvolvedores do Wammu (ou de outro *front-end*, se o Wammu se mostrar ineficiente) poderiam usar um script desses como base para adicionar monitoramento aos seus programas gráficos no futuro.

Dependendo do tipo de telefone (e acessórios) que o usuário possuir, o Gammu oferece um leque bastante grande de recursos. O comando `gammu -nokiamakecamerashoot` dispara a câmera digital de seu celular Nokia – caso ele possua uma – e tira uma fotografia. Para ligar o rádio, nada mais simples: `gammu -nokiatureradio`.

O programa trabalha melhor com telefones Nokia. Isso parece dever-se ao fato de ser baseado em um projeto já defunto chamado *Gnokii*, que se concentrava unicamente nos dispositivos dessa marca. No momento, o excelente e bem equipado programa em modo texto carece desespe-

radamente de um *front-end* gráfico à altura. O Gammu, não só por estar na linha de comando mas por ser realmente complexo, requer do usuário um conhecimento técnico por demais avançado. Embora o Wammu cumpra o papel de facilitador nas tarefas mais simples, outras de igual importância não são completamente satisfeitas – ou, pior, simplesmente não estão implementadas. Mesmo assim, ainda pode valer a pena testar ambas as ferramentas. Se o usuário for perseverante, o Gammu pode ajudá-lo a acessar praticamente todos os cantinhos do seu telefone celular. ■

INFORMAÇÕES

[1] Gammu: www.gammu.net

[2] Wammu: www.cihar.com/gammu/wammu

[3] Python Gammu :
www.cihar.com/gammu/python

[4] Lista de telefones reconhecidos pelo Gammu:
www.cihar.com/gammu/phonedb