

Conectando o Linux a seu telefone celular com Bluetooth

Telefone sem fio

É cada vez mais comum os telefones celulares saírem de fábrica com uma interface Bluetooth integrada, mesmo os mais baratos. Este artigo explica como acessar seus dispositivos Bluetooth usando o Linux.

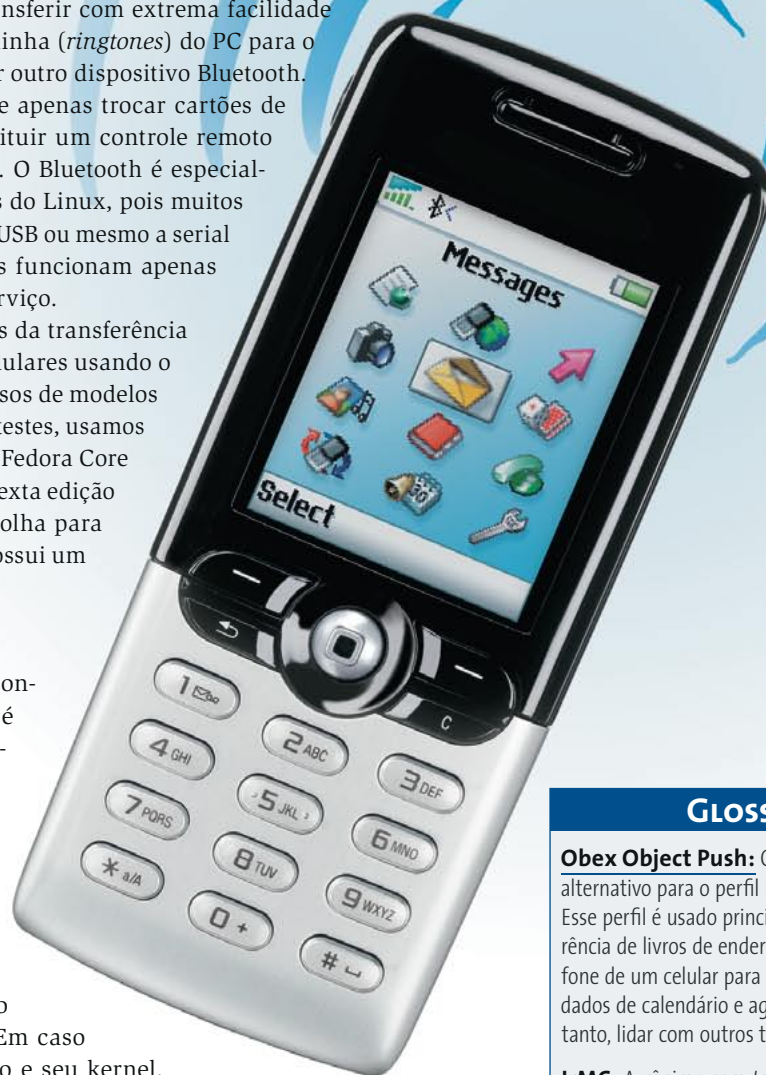
POR MARCEL HILZINGER

O Bluetooth tem potencial para se tornar um dos brinquedos mais populares dos próximos anos. Como o tráfego Bluetooth é transmitido diretamente de um dispositivo a outro, usar a tecnologia não custa nada. É possível transferir com extrema facilidade papéis de parede e toques de campainha (*ringtones*) do PC para o telefone celular e dele para qualquer outro dispositivo Bluetooth. O protocolo pode fazer mais do que apenas trocar cartões de visita e fotografias – dá para substituir um controle remoto infravermelho e mesmo um mouse. O Bluetooth é especialmente interessante para os usuários do Linux, pois muitos dispositivos que podem usar a porta USB ou mesmo a serial para se comunicar com o Windows funcionam apenas via Bluetooth se o Tux estiver de serviço.

Este artigo apresenta os princípios da transferência de arquivos entre PCs e telefones celulares usando o KDE. Mais adiante, explora os recursos de modelos específicos de telefones. Para nossos testes, usamos o SUSE Linux 9.1 e 9.2, bem como o Fedora Core 3. O SUSE Linux 9.2, distribuído na sexta edição da Linux Magazine, é uma boa escolha para iniciantes no Linux, já que o YaST possui um módulo específico para Bluetooth.

Alô? Quem está falando?

Para que seu aparelho Bluetooth converse com o Linux por telefone, é preciso que seu PC possua uma interface interna para Bluetooth ou, no mínimo, um “chaveiro” USB com adaptador Bluetooth. O kernel 2.6 do Linux reconhece nativamente interfaces Bluetooth ligadas via USB. Se seu kernel for da série 2.4, muitos adaptadores não vão funcionar ou, no mínimo, precisarão de um patch. Em caso de dúvida atualize sua distribuição e seu kernel. Depois de plugar o adaptador, o próximo passo é iniciar o subsistema Bluetooth (é preciso ser usuário *root* para isso) com o comando `/etc/init.d/bluetooth start`. Depois, digite `hcitool dev` para verificar se o Linux detectou corretamente seu adaptador Bluetooth. O resultado deve ser uma linha com o nome de dispositivo `hci0` e um endereço de hardware (por exemplo `00:10:C6:29:2E:13`).



GLOSSÁRIO

Obex Object Push: O *Obex Push* é o nome alternativo para o perfil Bluetooth *Object Push*. Esse perfil é usado principalmente para transferência de livros de endereços e números de telefone de um celular para outro, bem como de dados de calendário e agenda. Ele pode, entretanto, lidar com outros tipos de arquivos.

IrMC: Acrônimo para *Infrared Mobile Communications* (comunicação com dispositivos móveis por infravermelho). O Bluetooth usa uma especificação de transferência de arquivos originalmente projetada para conexões via infravermelho, que hoje estão caindo em desuso.

Agora que o subsistema Bluetooth do kernel está rodando, chame o ambiente Bluetooth do KDE digitando `kbluetoothd`. Já no Konqueror, digite a URL `bluetooth:/` para ver se o navegador do KDE consegue encontrar seu celular (ver **figura 1**). Antes de mais nada, habilite o Bluetooth em seu celular – acredite, isso vai poupar horas de seu trabalho e muita dor de cabeça – e deixe-o por perto. Para mais informações sobre como fazer isso em seu telefone, consulte a documentação do fabricante.



Figura 1: Além de si mesmo (*localhost*) o Konqueror vê dois celulares, um Pocket PC e outro PC de mesa.

Para enviar um arquivo para seu celular clique no ícone correspondente, rotulado como *K700i* em nosso exemplo. Com isso, obtemos uma lista dos serviços Bluetooth de que o celular dispõe (**figura 2**). Em nosso caso, isso significa clicar no ícone **Obex Object Push** e, em seguida, na opção **Open with "kbtobexclient"** na caixa de diálogo subsequente. Duas coisas podem acontecer:

- ⇒ O cliente *Bluetooth Obex Push Client* do KDE será chamado;
- ⇒ O sistema solicitará um PIN (*Personal Identification Number*) para conectar o PC ao celular.

Falando ao telefone

Antes que seu PC possa enviar qualquer coisa ao celular, é preciso, antes, conectar-se a ele. Esse processo, chamado de *pairing* (“emparelhamento”), é baseado em um código PIN compartilhado que, no Linux, é armazenado no arquivo `/etc/bluetooth/pin`. Como o arquivo, por padrão, está ajustado para um dispositivo chamado *BlueZ*, é preciso editar o arquivo antes de “emparelhar”. O arquivo deve conter uma única linha com um número à escolha do usuário, sem o prefixo *PIN:*. Se escrevermos o PIN do telefone celular, o programa definido como *pin_helper*, especificado em `/etc/bluetooth/hcid.conf`, verificará se os códigos combinam com o gravado em `/etc/bluetooth/pin`. Depois de “casar” os dois dispositivos com sucesso, o celular adiciona o PC à lista de dispositivos conhecidos.

O KDE dispõe de um *front-end* para a configuração dos PINs dos dispositivos; é preciso instalar o *KDE Bluetooth Framework*, o ambiente Bluetooth do KDE, para isso. O SUSE Linux, nas

Quadro 1: Configurando o Bluetooth

O sistema Bluetooth compreende um certo número de *daemons*, embora você normalmente precise apenas do `sdpd` e do `hcid`. O *Host Controller Interface Daemon* lida com as funções básicas do Bluetooth, enquanto o *Service Discovery Protocol Daemon* conversa com os outros parceiros de comunicação a respeito das habilidades do dispositivo Bluetooth instalado no PC. É possível iniciar esses *daemons* tanto manualmente como pelo script `/etc/init.d/bluetooth start`.

Para iniciar o subsistema Bluetooth automaticamente após o boot, digite (como *root*) o comando: `insserv bluetooth` no SUSE. Para o Fedora Core, digite `chkconfig --add bluetooth` ou inicie o editor gráfico pelo menu *Applications | System settings | Server settings | Services*. Os usuários do SUSE poderão também usar o *Runlevel Editor* no YaST.

O arquivo central de configuração para o sistema Bluetooth é chamado de `/etc/bluetooth/hcid.conf`. A primeira coisa a fazer é mudar o nome (*name*) na opção `Default settings for HCI devices`, pois o nome padrão para a maioria das distribuições é *BlueZ*.

Alguns programas não conhecem a tecnologia Bluetooth. Para eles, será preciso emular uma porta serial usando o `rfcomm`. Isso permite que enviemos comandos de modem ao celular para verificar o estado da bateria ou enviar mensagens curtas. O comando para isso é `rfcomm bind 0 <endereço_bluetooth>`; substitua `<endereço_bluetooth>` pelo endereço de hardware de seu dispositivo Bluetooth. Para que qualquer usuário possa acessar o dispositivo `rfcomm0`, é preciso modificar os privilégios de serviço. Novamente como *root*, digite `chmod 666 /dev/rfcomm0`.

Como as distribuições Linux criam os arquivos em `/dev` dinamicamente (veja matéria sobre Hotplug na página 50), será preciso rodar esse comando após cada `rfcomm bind`. Se você precisar alterar os privilégios de acesso de forma permanente, adicione a linha a seguir no final do arquivo `/etc/udev/udev.permissions` (no SUSE Linux 9.1) ou em `/etc/udev/permissions.d/50-udev.permissions` (no Fedora Core e no SUSE Linux 9.2):

```
rfcomm*:root:root:666
```

O comando `rfcomm` emitido sem parâmetros mostra uma lista dos dispositivos casados. Para finalizar uma conexão existente digite `rfcomm release /dev/rfcomm0`. Alguns telefones móveis em nosso teste não foram capazes de enviar arquivos enquanto estavam casados com a interface `/dev/rfcomm0`.

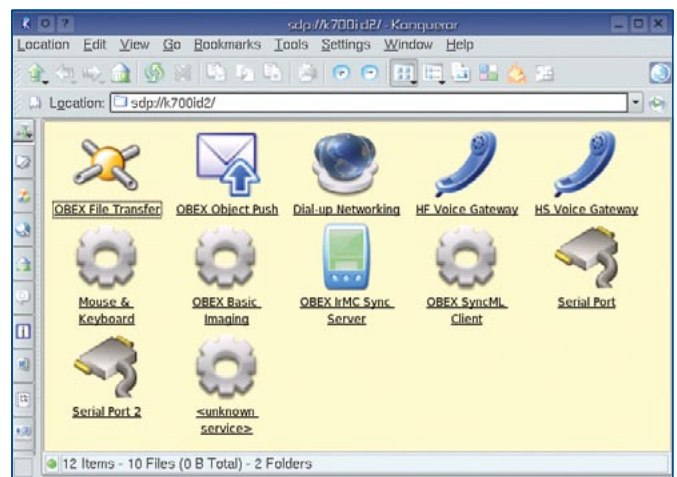


Figura 2: O K700i possui um impressionante cardápio de serviços. Escolha o *Obex Object Push* para enviar arquivos ao celular.

versões 9.1 e 9.2, instala os pacotes necessários por padrão. O site oficial do *KDE Bluetooth Framework* [1] também possui pacotes pré-compilados para outras distribuições. Mude o programa `pin_helper` para `/opt/kde3/lib/kdebluetooth/kbluepin`. Dessa forma teremos o programa gráfico do KDE nos ajudando com nossos PINs. O *daemon* `kdebluetooth` perguntará qual o caminho ao ser executado. Não é preciso um PIN estático com o *KDE Pinhelper*. Digite o PIN na caixa de diálogo (veja a **figura 3**). ➔

Transferência de arquivos

Depois de “casar” os dispositivos, o cliente *Obex Push* do KDE será chamado à luta (ver [figura 4](#)). Na parte superior da janela estão os arquivos que podem ser transferidos. Arraste os escolhidos para o ícone *File to send*. O seletor de dispositivos deve indicar seu celular. Se o *Kbtobexclient* não conseguir encontrar um dispositivo, clique em *Search* para fazer uma busca. O botão *Send* transfere os arquivos para o celular. Dependendo do modelo, será preciso aceitar, no celular, a transferência. O tipo de arquivos que um dispositivo aceita depende do conjunto de recursos de que dispõe. Os tipos a seguir funcionaram em todos os celulares que testamos:

- ⇒ Cartões de visita eletrônicos (*.vcf)
- ⇒ Compromissos e calendário (*.ics e *.vcs)
- ⇒ Imagens (*.jpg)
- ⇒ Toques de campainha MIDI (*.mid)

Transferir arquivos de seu celular para o PC é, também, tarefa fácil. No menu apropriado do aparelho, selecione a opção de transferência. Seu telefone deve automaticamente procurar por outros dispositivos Bluetooth. Seu PC será o dispositivo cujo nome você registrou em `/etc/bluetooth/hcid.conf`. Depois, clique no botão *Send*. Praticamente todos os telefones que testamos possuem um item no menu chamado *Send | Via Bluetooth*. O Motorola V600 e o MDA III são as raras exceções. No Motorola, a opção no menu pode ser *Move* ou *Copy*, dependendo do contexto. O PDA com o Windows CE não possui um menu e espera que o usuário selecione os arquivos a enviar pelo gerenciador Bluetooth integrado – um processo complicado que descreveremos mais adiante.

Se o *daemon* do KDE Bluetooth estiver rodando, uma janela deverá pipocar em sua área de trabalho indicando que uma transferência de arquivos está em curso. A opção *Future policy for this device* (comportamento futuro para este dispositivo)

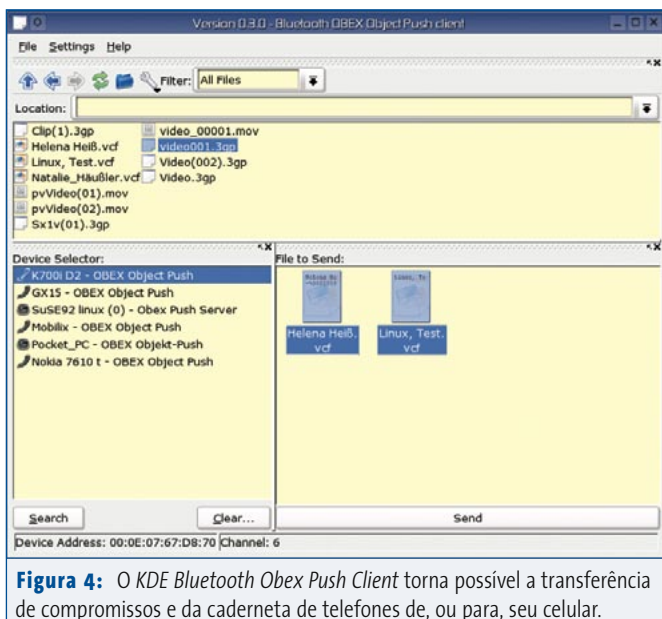


Figura 4: O KDE Bluetooth Obex Push Client torna possível a transferência de compromissos e da caderneta de telefones de, ou para, seu celular.

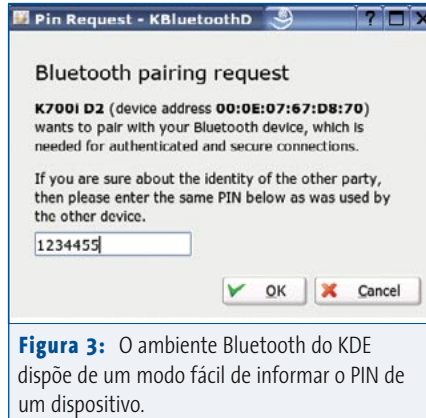


Figura 3: O ambiente Bluetooth do KDE dispõe de um modo fácil de informar o PIN de um dispositivo.

permite escolher entre automaticamente aceitar ou rejeitar arquivos desse dispositivo no futuro. Ao aceitar a transferência, a janela deve mostrar *Incoming File Transfer* (transferindo arquivos do celular). É possível clicar no arquivo para abri-lo ou escolher *Save* para gravá-lo no disco rígido.

Os usuários do Fedora Core precisam dos pacotes *gnome-bluetooth* e *gnome-vfs-extras* para trocar arquivos com seus telefones celulares. Se o Fedora se recusar a aceitar dados do telefone, é necessário rodar o programa *gnome-obex-server*

primeiro. Se o programa *Gnome Bluetooth* estiver instalado, novamente uma janela irá desmontar na tela e perguntar se aceita ou não a transferência. O Fedora grava os arquivos recebidos em seu diretório pessoal (`/home`).

Cuidado com as armadilhas do SUSE Linux 9.2! Quando os dispositivos são casados, nenhuma caixa de diálogo é mostrada. Em vez disso, os arquivos são armazenados automaticamente em `/var/lib/bluetooth`. Isso quer dizer que qualquer um que saiba seu PIN pode mandar arquivos para a sua máquina sem que você perceba – uma falha de segurança gravíssima!

Competição azul

Em nosso laboratório, testamos nada menos do que oito telefones Bluetooth (nem todos disponíveis no Brasil, veja a [tabela 1](#)). Esses aparelhos foram gentilmente colocados à nossa disposição pelas operadoras de celular. Há três grupos de aparelhos. O primeiro possui um mostrador bem pequeno e um sistema operacional proprietário. O Sony Ericsson modelos T610 e K700i, o Motorola V600 e o Sharp GX15 estão nesse grupo.










O segundo grupo é composto pelos celulares com o sistema operacional Symbian. O grupo inclui os Nokia 6600 e 7610 e o Siemens SX1. O terceiro grupo é formado por um só dispositivo da operadora alemã T-Online, o MDA III com o sistema operacional Windows CE.

Com a notável exceção do Nokia 7610, que possui uma câmera de um megapixel (1152x864), todos esses telefones possuem uma câmera VGA (640x480) que pode tirar fotos e capturar filmes em formato 3GP – na verdade, um formato baseado no Quicktime, que pode ser visto no Linux com o Realplayer. Se você não tiver o Realplayer instalado, mude a extensão do arquivo de `*.3gp` para `*.mov`, permitindo assim que outros reprodutores de mídia possam mostrar o filme, entretanto sem som (pelo menos em nossos testes). O Motorola V600 não possui o recurso de gravação de vídeo.

Sincronia IrMC

Os aparelhos que mais nos impressionaram no primeiro grupo foram o Sony Ericsson T610 e o K700i. Ambos são bastante fáceis de usar e proporcionam aos usuários do Linux as mesmas conveniências que os do Windows esperam de um celular dessa categoria. Esses dois telefones foram os dois únicos em nosso laboratório a reconhecer o protocolo **IrMC**, que sincroniza os dados entre o dispositivo móvel e os aplicativos *Kaddressbook* e *Korganizer* no KDE. O programa *Multisync* deveria fazer a sin-

Tabela 1: Visão geral dos telefones móveis

									
Modelo	T610	k700i	GX15	V600	SX1	6600	7610	MDA III	
Fabricante	Sony Ericsson	Sony Ericsson	Sharp	Motorola	Siemens	Nokia	Nokia	HDC	
Operadora	E-Plus	Vodafone D2	Vodafone D2	O2	T-Mobile	T-Mobile	T-Mobile	T-Mobile	
Sistema operacional	proprietário	proprietário	proprietário	proprietário	Symbian - Series60	Symbian - Series60	Symbian - Series60	Windows CE	
Memória interna	2 MB	32 MB	2 MB	5 MB	24 MB	6 MB	8 MB	128 MB	
Transferência de arquivos por Obex	✓	✓	x	✓	✓	✓	✓	✓	
Controle remoto Bluetooth	✓	✓	x	x	x	x	x	x	
Controle remoto Bemused	x	x	x	x	✓	✓	✓	x	
Sincronia IrMC	✓	✓	x	x	x	x	x	x	
Sincronia SyncML	x	✓	x	x	✓	✓	✓	x	
Kmobiletools(1)	✓	✓	x	x	✓	✓	✓	x	
Reprodutor Ogg	x	x	x	x	✓	✓	✓	✓	
Cliente SSH	x	x	x	x	✓	✓	✓	✓	
Preço só do aparelho ²	R\$ 1.800,00	R\$ 1.699,00	Não disponível	R\$ 1.299,00	Não disponível	R\$ 1.499,00	R\$ 2.499,00	Não disponível	
Preço pré-pago ²	R\$ 599,00	R\$ 1.549,00	Não disponível	R\$ 1.349,00	Não disponível	R\$ 1.199,00	R\$ 2.199,00	Não disponível	
Preço pós-pago ²	R\$ 449,00	R\$ 1.399,00	Não disponível	R\$ 1.199,00	Não disponível	R\$ 999,00	R\$ 1.999,00	Não disponível	

¹ ✓ significa que pelo menos o mostrador do estado da bateria funcionou. Alguns dispositivos permitem o envio de mensagens SMS e a leitura do livro de endereços.

² preço médio no mercado brasileiro, entre as operadoras Oi, TIM e Claro.

crônica com o *Evolution*, cliente de email do Gnome, mas não há plugin disponível para a versão 2.0 do programa. É por isso (e apenas por isso) que daremos destaque ao KDE nesta edição, com o uso do aplicativo *Kitchensync* (um trocadilho infame com *Kitchen Sink*, a pia da cozinha).

Para permitir que o KDE Bluetooth e o *Kitchensync* conversem entre si, certifique-se de que o *KDE Bluetooth Framework* foi compilado com a opção *Irmcsyncconnector*. O SUSE Linux 9.2 possui pacotes RPM prontos com esse recurso ativado. De fato, o *Kitchensync* funcionou sem “grilos” no SUSE Linux 9.2 de nosso laboratório. De qualquer modo, não importa o que você faça, crie cópias de segurança de seus livros de endereços no celular e no KDE antes de começar a brincar com essas coisas.

Para iniciar o *Kitchensync*, pressione **[Alt]+[F2]** para abrir a caixa de diálogo *Executar comando...* e digite

kitchensync. Agora clique em *Settings | Configure KitchenSync* e selecione a opção *connector* em *Resources*. Clique no botão *Add...* e adicione os conectores *Irmcsync connector* e *Local connector*.

Como recurso *Irmcsync*, selecione o seu aparelho de celular em *Bluetooth Device*. Na porção inferior da janela, especifique se quer sincronizar seu livro de endereços ou a agenda. Cada conector pode ser usado em apenas uma direção (ver *figura*

5a). Para evitar que o *Kitchensync* escreva por cima da caderneta de telefones em seu celular, ative a opção *Read only* (apenas leitura). Se seu telefone não estiver na lista de opções, provavelmente ele não permite sincronia por *IrMC* (ver *tabela 1*) ou não está configurado como visível.

No conector local (ver *figura 5b*), especifique um livro de endereços e um arquivo de agenda. A maneira mais fácil é clicar em *Select From Existing*

Resources (selecionar a partir dos recursos disponíveis), ordenando assim que o *Kitchensync* selecione automaticamente o livro de endereços do KDE e a agenda do *Korganizer*. Novamente, ative a opção *Read-only* (apenas leitura) para evitar que o *Kitchensync* faça alguma bobagem e danifique sua agenda de telefones local. Depois de adicionar os conectores, clique em *Edit...* (editar) para definir os parâmetros apropriados. ➔



Figura 5a: Seu telefone deve estar listado como um recurso *Irmcsync*.

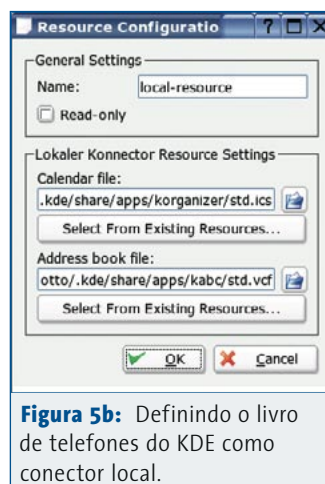


Figura 5b: Definindo o livro de telefones do KDE como conector local.



Figura 6: Configurando o livro de endereços do KDE pelo conector local.

Neste ponto, selecione **Settings | Configure Profiles** para especificar os perfis que o *Kitchensync* deve considerar. No SUSE Linux 9.2 há dois perfis por padrão: *Syncing* (sincronização) e *Restore Backup* (restaurar cópia de segurança). Se não houver nenhum perfil, clique em **Add...**, digite o nome desejado e defina os seguintes **Parts** (componentes):

- *Overview* (visão geral)
- *Connector Backup* (cópia de segurança do conector)
- *Synchronizer* (sincronizador)

Experimente com os outros *parts* e veja o que eles podem fazer por você. Os perfis aparecem na janela principal do *Kitchensync* na caixa de seleção abaixo dos menus. Ao selecionar o perfil *Syncing*, os ícones à esquerda da janela informarão que há três componentes: *Overview*, *Connector Backup* e *Synchronizer*. O perfil *Restore Backup* possui apenas um componente chamado *Connector Backup*. Para sincronização dos dados entre o celular e o PC, o perfil apropriado é

Quadro 2: Ferramentas úteis para Bluetooth

Kmobiletools: Lê o estado da bateria e a qualidade do sinal, consulta o livro de telefones, lê e envia mensagens SMS – mas suporta apenas em alguns aparelhos. Só funciona se os telefones forem casados via `rfcomm`. A página oficial é <http://kmobiletools.berlios.de/>

Multisync: Um programa utilíssimo para criar uma cópia de segurança da caderneta de telefones de seu celular ou para sincronizá-la com o livro de endereços do KDE ou Evolution. Graças à arquitetura de plugins, o *Multisync* pode usar várias fontes de dados. Por exemplo, é possível trocar dados com servidores LDAP e SyncML.

Kbthandfree: Parte do *KDE Bluetooth Framework*. Permite a operação sem usar as mãos usando um

Syncing. Clique no ícone próximo à caixa de seleção (ver **figura 6**). Dependendo do tamanho do caderninho de telefones, pode levar alguns minutos para o *Kitchensync* completar a primeira sincronização. Da próxima vez, entretanto, as coisas serão muito mais rápidas.

Se o programa detectar um item que seja semelhante em ambas as cadernetas, pipoca na tela uma caixa de diálogo semelhante à da **figura 7**. Isso permite que o usuário escolha qual delas o *Kitchensync* deve considerar. Infelizmente, o SUSE Linux 9.2 não escreve nada no texto do botão correspondente ao celular; apesar disso, o botão funciona. Se encontrar alguma dificuldade ao sincronizar algum dispositivo, adicione um componente de depuração (*debugger*) a seu perfil. Assim, é possível navegar por todas as etapas na comunicação e ver qual delas está perturbando o ambiente.

Controle remoto

Ficamos fascinados pelo controle remoto *Bluemote* [2], que permite controlar remotamente o reproduutor de mídia XMMS e uma apresentação do OpenOffice.org usando o telefone celular. Antes de começar, é preciso improvisar uma conexão serial virtual para seu aparelho usando o comando `rfcomm bind 0 <endereço_bluetooth>` (ver **quadro 1**). A ferramenta `hcitool scan` informa o endereço de hardware do seu telefone:

```
kim:~ # hcitool scan
Scanning ...
00:0E:07:37:60:03 T610
```

fone de ouvido Bluetooth. Para usá-lo, execute o comando `kbthandfree`, selecione seu fone na janela *Bluetooth Service Selection* (o telefone tem que estar visível para que isso funcione) e ative a opção **Voice over Handsfree**. Para atender a qualquer telefonema basta, a partir de agora, clicar em **Accept** quando o telefone tocar. Para fazer uma chamada, digite o número e clique em **Dial**. Use o menu **Settings | Configure Bluetooth Handsfree** para que o programa grave todas as conversas em um arquivo de áudio no formato Ogg Vorbis.

Gnome Phonemanager: Minúscula ferramenta para enviar e receber mensagens via SMS. Ao receber uma mensagem SMS, uma janela com a mensagem aparece na área de trabalho. Funcionou no SUSE Linux 9.2 com todos os dispositivos que testamos, exceto o Motorola V600 e o MDA III. O site é <http://usefulinc.com/software/phonemgr/>

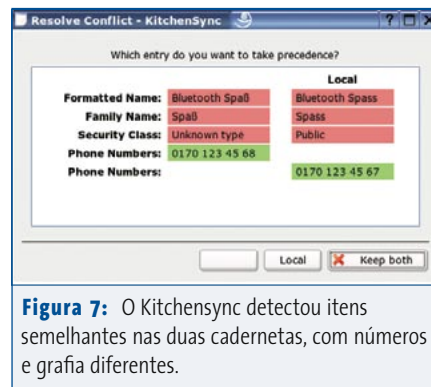


Figura 7: O *Kitchensync* detectou itens semelhantes nas duas cadernetas, com números e grafia diferentes.

Agora digite `bluemote` e verifique se a palavra *Bluemote* aparece no mostrador do seu celular. Selecione **Connections | Tools | PC Remote** para ver os itens individuais de controle remoto. O programa `aumix-minimal` é necessário para ajustar o volume do reproduutor de mídia usando o controle de *Volume*.

De todos os dispositivos testados, a tecnologia *Bluemote* funcionou apenas nos telefones Sony Ericsson. Não há ferramentas Linux que desempenhem as mesmas funções no Sharp GX 15 e no Motorola V600; de qualquer maneira, esses fones possuem uma gama de recursos muito limitada.

Os anos 60...

O segundo grupo de candidatos de teste é formado pelos aparelhos Nokia 6600, Nokia 7610 e Siemens SX1, todos com o sistema operacional *Symbian Series 60*. As estrelas do céu não bastam para contar o número de programas gratuitos para o *Symbian* disponíveis na Internet, o que significa que esses três telefones funcionam magicamente no Linux. Isso da parte deles, pois do lado do Linux não há, ainda, um programa confiável de sincronia para a tecnologia usada por eles, baseada no novíssimo protocolo *SyncML*. O *Multisync* [3] é um cliente Linux para o *SyncML*, porém ele reconhece apenas servidores HTTP – o sincronismo direto com dispositivos Bluetooth está fora de questão. Para os aventureiros, há um tutorial [4] que descreve como configurar uma conexão IP/Bluetooth com telefones baseados no sistema operacional *Symbian*, mas é apenas um improviso para enganar o *Multisync* e permitir o sincronismo com ele. Mas não desanime: há desenvolvedores suando para trazer à luz um conector *SyncML* para o *Kitchensync* [5], embora não tenhamos tido oportunidade de testá-lo enquanto escrevíamos o artigo.

Há aproximadamente 15.000 programas para telefones *Symbian Series 60*, incluindo um reprodutor de arquivos Ogg-Vorbis [6], um cliente SSH [7] e um servidor (!) NFS, provido de um cliente apropriado [8]. O reprodutor Ogg e o cliente SSH rodam sem alterações, bastando transferir o arquivo SIS para o telefone. Acessar os arquivos de um telefone *Series 60* via NFS requer um pouco mais de trabalho.

Instale em seu Linux o pacote P3NFS (no SUSE é `p3nfs-5.16-1.i386.rpm`) e transfira o arquivo SIS disponível em `/usr/share/doc/p3nfs-5.16/` para seu telefone. Crie uma conexão `rftcomm` com o comando `rftcomm bind 0 <endereço_bluetooth> 11` (para mais detalhes veja o **quadro 1**). Agora inicie o cliente e o servidor NFS no telefone digitando `p3nfsd -series60 -tty /dev/rftcomm0` no Linux. O `daemon` NFS tentará montar seu telefone em `/mnt/psion`. Se preferir usar outro ponto de montagem (por exemplo, porque não há um diretório `/mnt/psion` em seu computador) estipule a opção `-dir /um/diretório/à/escolha` ao emitir o comando. O diretório deve mostrar os cartões de memória em seu telefone como “drives” chamados A:, C:

e E:. Para desativar o servidor, digite o seguinte comando – obviamente, com o ponto de montagem correto:

```
ls /mnt/psion/exit
```

...e seus brinquedos!

O *Bemused* [9] faz para os telefones com Symbian o mesmo que o *Bluemote* fez para os modelos Sony Ericsson. Esse programa controla remotamente o XMMS e o Noatun. O *Bemused* concentra-se nos reprodutores de mídia, já que não pode controlar o mouse. O programa procura por arquivos de áudio e vídeo em seu PC e reproduz os selecionados pelo usuário. Você pode também baixar algumas das faixas para seu telefone ou cartão de memória, desde que tenha espaço livre suficiente.

O **quadro 2** mostra algumas das ferramentas que podem ser usadas com os telefones *Series 60* e outros. Também recomendamos dar uma olhada no CD que veio junto com seu telefone – deve haver muitas coisas divertidas por lá. Os CDs dos fabricantes normalmente trazem jogos e programas de demonstração que podem ser instalados via Bluetooth.

Obex Direct Access

Os telefones *Series60* permitem acesso direto a seus arquivos sem esse improviso com NFS. Simplesmente digite `bluetooth:/` em seu Konqueror, selecione o telefone desejado (se houver mais de um) e clique em `OBEX File Transfer` (ver **figura 2**).

Dependendo do dispositivo, é possível arrastar e soltar arquivos entre o PC e o telefone dentro do Konqueror. Ao contrário do servidor P3NFS, é possível fazer transferência simultânea de diversos arquivos. A transferência de arquivos por *Obex* é possível em todos os aparelhos, exceto o GX15, embora o K700i e o V600 tenham nos dado algum trabalho. Quando tentamos contatar o dispositivo, o Konqueror solicitou autenticação. Depois de cancelar a mensagem, pudemos transferir arquivos para o telefone, mas não conseguimos vê-los lá. Isso se deve a um erro no Kdebluetooth, já consertado na versão em desenvolvimento. Quando baixamos e instalamos uma versão mais recente do Kdebluetooth do site dos desenvolvedores, conseguimos ver as pastas no telefone: *Others*, *Images*, *Displayprofile*, *Sounds* e *Videos*. ➡

Quadro 3: O futuro do Bluetooth

Há uma nova especificação surgindo no horizonte tecnológico. O futuro do Bluetooth chama-se 2.0 + EDR. A nova especificação, promovida pelo Bluetooth SIG (*Bluetooth Special Interest Group* ou grupo especial de interesse na tecnologia Bluetooth), objetiva incrementar a velocidade de transmissão de dados para dispositivos Bluetooth – dos atuais 80 kbit/s para impressionantes 240 kbit/s. Isso coloca o Bluetooth em condições de permitir a transmissão de programas de áudio e vídeo. Em outras palavras, vai ser possível usar seu fone de ouvido Bluetooth para ouvir CDs de áudio e arquivos MP3 de alta qualidade. Há outras aplicações como, por exemplo, permitir que estabelecimentos comerciais possam imprimir em papel fotográfico as fotos digitais que seus clientes capturaram em seus telefones com câmera. Tudo via Bluetooth, sem sequer tirar o celular do bolso...

A nova especificação também permite associações múltiplas. Os adaptadores atuais permitem um máximo de oito nós em uma rede Bluetooth. Como a tecnologia está se tornando cada vez mais comum, com teclados, mouses, câmeras digitais e outros periféricos malucos para PC, esse limite é restritivo demais para aplicações futuras. O novo limite projetado é de 256 dispositivos.

Finalmente, o Bluetooth SIG está trabalhando para melhorar a segurança das conexões e o alcance dos adaptadores. Há alguns adaptadores USB para Bluetooth que permitem distâncias de até 100 metros em visada direta. A idéia é fazer dessa distância o padrão mínimo aceitável num futuro próximo.

O telefone celular O2 mostra pastas chamadas *audio*, *picture* e *video* quando clicamos no *OBEX File Transfer*; um segundo clique abre uma das pastas. Entretanto, se você quiser transferir um arquivo MIDI para usar como toque de campainha, o diretório apropriado não é *audio*. Em vez disso, arraste e solte o arquivo para a janela com os três diretórios. O aparelho grava o arquivo no lugar certo automaticamente, baseado na extensão de seu nome.

O megafone

O luxo proibitivo que testamos carrega em sua carcaça uma alma Windows CE. Com seus 128 MBytes de RAM, adaptador para rede sem fio WiFi e uma tela LCD de 240 por 320 pixels, o MDA III está mais para computador de mão do que para telefone celular. Na primeira vez que ligamos o bicho, um assistente do Windows permite ajustar a tela. Depois, outro assistente ensina a usar as funções de toque da tela *touchscreen*. Depois disso, o Windows instala alguns arquivos e cria sua área de trabalho. Esses procedimentos roubam aproximadamente três minutos do seu tempo. Como a RAM interna do MDA III precisa de energia permanentemente, é preciso repetir todos esses passos a cada vez que, por algum motivo, a energia acabe – na troca de baterias, por exemplo. O Windows simplesmente esquece as configurações, incluindo as da interface Bluetooth.

Enquanto outros celulares possuem um menu chamado *Send via Bluetooth* (enviar via Bluetooth), o Windows CE

nada mostra – nem esse, nem qualquer outro menu Bluetooth. Antes, é preciso rodar o painel de controle Bluetooth selecionando *Start | Programs | Bluetooth Manager*. Seleccione depois *Connections | New | Transfer files* e escolha os arquivos a transferir. Em nossos testes, na maioria das vezes isso falhou. Em vez disso, o Windows diz que o computador selecionado não reconhece o aparelho. Se isso ocorrer, verifique se a opção *Password key required* está ligada em *Bluetooth | Services | File transfer*; se estiver, desative-a.

Infelizmente o dispositivo não reconhece comandos AT, como os que usamos com o Kmobiletools para verificar o estado da bateria ou para enviar mensagens. E há uma oferta bem mirradinha de programas gratuitos para o MDA III. Pelo menos encontramos um reproduzidor de arquivos Ogg-Vorbis em [10] e um cliente SSH em [11]. Procuramos incessantemente, mas não encontramos nada parecido com os controles remotos *Bemused* e *Bluemote* para o MDA III.

A transferência de arquivos por *Obex*, entretanto, funcionou maravilhosamente bem. Foi possível ler e escrever nas pastas compartilhadas do telefone (normalmente, apenas *Documents*). O projeto *SyncCE* em [12] também permite acesso ao protocolo proprietário de sincronização da Microsoft. Há plugins para o *Multisync* e o *Kitchensync*, portanto é possível usá-los com o Evolution e o KDE.

Graças ao cliente IrMC, o Linux pode trabalhar com todos os recursos dos aparelhos Sony Ericsson T610 e K700i.

Aparelhos com o Symbian são úteis para os administradores ávidos por novidades, embora exijam alguma atenção manual antes que se possa sincronizar por eles. Pelo lado positivo, o cliente de NFS permite até que se faça scripts!

Em nossos testes, foi difícil dizer algo de bom ou de ruim a respeito do Motorola V600 ou do Sharp GX 15. A despeito de sua extensa lista de recursos, o caríssimo MDA III não convenceu. A função *Send via Bluetooth*, algo básico em um telefone com essa tecnologia, não está presente. Pior: a solução para isso torna o uso do celular com Windows complicado demais para o usuário médio. Sem contar a pífia duração da bateria, de meras nove horas.

Durante a redação deste artigo, consegui finalmente sincronizar meu próprio telefone, um já ancião Ericsson r520, usando o *Kitchensync*. Até o controle remoto funcionou: consegui controlar o *Kaffeine* usando o *Bluemote*. Portanto, mesmo que seu telefone não tenha sido explicitamente mencionado neste artigo, é possível que as técnicas mostradas sejam bastante úteis aos felizes proprietários de telefones Bluetooth. ■

INFORMAÇÕES

[1] KDE Bluetooth Framework:
<http://kde-bluetooth.sf.net>

[2] Controle remoto Bluemote:
<http://www.geocities.com/saravkrish/progs/bluemote/>

[3] Multisync: <http://multisync.sf.net>

[4] IP sobre Bluetooth para Symbian Series60:
<http://www.unix-ag.uni-kl.de/~leonard/linux-n6600-howto.html>

[5] Conector SyncML para Kitchensync:
<http://www.borowka.net/~maciek/ksyncml/>

[6] Reproduzidor Ogg para Series60:
<http://symbianoggplay.sf.net>

[7] Cliente de SSH para Series60:
<http://s2putty.sf.net/>

[8] Cliente de NFS para Series60:
<http://www.koeniglich.de/p3nfs.html>

[9] Controle remoto Bemused:
<http://bemused.sf.net>

[10] Reproduzidor de Ogg para o Windows CE:
<http://www.sto-helit.de/freeware/pocketpc/mortplayer/en/index.pl>

[11] Cliente de SSH para o Windows CE:
<http://pocketputty.duxy.net>

[12] Projeto SynCE: <http://synce.sf.net/synce/>