UNIVERSIDADE DE SÃO PAULO – ESCOLA POLITÉCNICA

DEPARTAMENTO DE ENGENHARIA MECATRÔNICA

nota final
6,5 (seis e meio)

SISTEMA DE AUTOMAÇÃO PREDIAL COM SUPERVISÃO POR TELEFONIA

JOSÉ ADRIANO MEIRELES PARDI

SÃO PAULO

2005

JOSÉ ADRIANO MEIRELES PARDI

SISTEMA DE AUTOMAÇÃO PREDIAL COM SUPERVISÃO POR TELEFONIA

Monografia apresentada à Escola Politécnica da Universidade de São Paulo para a obtenção da graduação em Engenharia Mecânica de Automação e Sistemas

Orientador:

Prof. Dr. José Reinaldo Silva

São Paulo

2005

RESUMO

Com o desenvolvimento progressivo das tecnologias de comunicação à distância, é crescente a classe de sistemas que se valem das mídias digitais para estabelecer conexões de voz entre dois pontos, destacando-se a tecnologia de Voz Sobre IP, ou VoIP. Os sistemas baseados nessa técnica de transmissão de dados em tempo real têm sido cada vez mais utilizados como alternativa aos meios tradicionais de comunicação, como a telefonia, especialmente para chamadas a longa distância. Por outro lado, a telefonia tradicional sofreu verdadeira revolução quando a ela se uniu o computador, produzindo os chamados sistemas de IVR (Interactive Voice Response), que possibilitam a interação remota de um usuário com o meio computacional através da reprodução, pelo sistema, de mensagens pré-gravadas. O presente trabalho, fazendo uso dessas duas tecnologias (IVR e VoIP), consiste num PBX (Private Branch Exchange) digital em que a comunicação entre os ramais em si, bem como entre cada ramal e o servidor, dá-se através do mecanismo de Voz Sobre IP. Adicionalmente, construiu-se, acoplado ao PBX, um sistema automático de monitoração que, ao detectar movimento no local monitorado (o que é feito usando-se uma webcam ordinária e seu software específico), aciona o servidor central do PBX que efetua automaticamente um telefonema para o usuário avisando-o da detecção de intrusão. Tanto o gerenciamento do PBX quanto o sistema de IVR foram feitos por programação de um software de código aberto, o Asterisk, que funciona no sistema operacional Linux. Objetivou-se, com isso, integrar a telefonia à automação predial, tirando proveito das vantagens dos softwares de código aberto, as quais refletiram-se para o sistema: pouca exigência em termos de hardware específico, otimização da capacidade de processamento dos computadores, baixo custo e alta flexibilidade em termos de acréscimo de novas funcionalidades. Como conclusão, verificou-se a excelente adaptabilidade dos sistemas baseados no Asterisk aos dispositivos de automação predial.

Palavras-chave: Automação Predial. Telefonia. Voz Sobre IP. Linux. Asterisk.

ABSTRACT

The development of the communications technology brings about a variety of systems which make use of the digital media in order to establish real-time voice connections between two points. This is the goal of the Voice Over IP (VoIP) technology. Due to their flexibility (run across the Internet), high performance and low cost, VoIP connections are being increasingly used in substitution of traditional telephony systems, especially for long-distance calls. In the other hand, telephony systems have been revolutionized by the use of computer-based technology, which provides the possibility of interaction between a remotely-situated user and an automatic computer-driven system, through the use of pre-recorded audio messages played by the system and the pressing of digits by the user in his telephone device. Such systems are called IVR (Interactive Voice Response) systems. In the present work, both technologies (VoIP and IVR) were used in order to build a digital PBX (Private Branch Exchange) system whose extensions communicate to each other via VoIP (using the SIP protocol). Attached to the PBX is an automatic monitoring apparatus that works by motion-detection. A common webcam, which is arbitrarily connected to the same PC that runs one of the extensions of the PBX, captures images of the place that is being monitored. Every time motion is detected by a proper software (part of the webcam package), a file is sent to the PBX server, causing the software that runs in this machine to make an automatic call to the user, warning him of the intrusion by playing pre-recorded audio messages. Additionally, an IVR menu is provided to the user, allowing him to enable/disable automatic warnings, as well as to retrieve information (date ant time) about the registered invasions. The software that controls the PBX and makes the automatic telephone calls was developed using open source platforms (including the Asterisk software), into Linux environment, resulting in a fully-functional system with high flexibility and low cost, using nothing more than 3 common network-connected PC's and a single card produced by Digium Inc. to connect the PC to the ordinary telephone line.

Keywords: Telephony. Interactive Voice Response. Voice Over IP. Linux. Asterisk.

Índice

1. Introdução

A evolução tecnológica nas áreas da comunicação que se observa nos últimos anos, em especial a tecnologia de telefonia, possibilita a concepção de uma gama de aplicações que vão além do objetivo clássico desses sistemas de comunicação, que é permitir o diálogo entre duas pessoas situadas em locais diferentes.

A implementação de microprocessadores poderosos em aparelhos de telefonia celular, aliada à extensão do sinal de telefonia celular digital para a conexão à Internet, vem transformando aparelhos telefônicos em computadores, com toda a variedade e o potencial para realizar as mais diversas tarefas, com as vantagens da extrema praticidade e do custo cada vez mais baixo.

Além disso, o mundo do software de código aberto (*open source software*) cresce num ritmo acelerado, colocando à disposição dos usuários e programadores uma variedade de ferramentas computacionais progressivamente mais avançadas que, muitas vezes, superam com larga vantagem a eficiência dos softwares comerciais correspondentes.

Neste trabalho, foi realizada a junção de um sistema de automação predial com o sistema de telefonia, criando um sistema automático de monitoração que possibilita ao usuário, por meio de uma simples ligação telefônica, conectar-se ao servidor e realizar com ele uma troca de informações, através do envio de comandos ao sistema de controle e da recepção de dados de supervisão fornecidos pelo mesmo.

A principal característica constitutiva deste projeto é que toda a programação do dispositivo computacional que envolve a interface com a telefonia foi feita usando software de código aberto, em ambiente Linux, exigindo um mínimo de hardware e capacidade de processamento. Com isso, oferecem-se às empresas todas as vantagens inerentes ao uso dessa plataforma, como:

- Estrutura altamente flexível, que via de regra pode ser implementada nas bases de hardware e software pré-existentes sem a necessidade de grandes modificações;
- Custos reduzidos devido à ausência de licenças dos softwares de código aberto;
- Facilidade para futuras implementações / acréscimo de novas funcionalidades, dada a arquitetura aberta do sistema.

2. Funcionamento Básico do Sistema

O sistema consiste em um PBX (*Private Branch Exchange*) digital[1] instalado no Design-Lab, cuja central é um computador rodando o Asterisk[2].

O servidor recebe dados do subsistema de monitoração (detecção de movimento) do local automatizado. Assim, quando ativado, o subsistema de monitoração manterá uma câmera do tipo webcam, num dos terminais do PBX, colhendo imagens do local monitorado. Havendo uma intrusão, o software[3] detectará a mudança na imagem capturada e acionará a central do PBX, que efetuará um telefonema para o usuário avisando-o de que houve detecção de movimento (*intrusão*).

Adicionalmente, as imagens capturadas, flagrando a intrusão, são enviadas para o e-mail cadastrado do usuário, possibilitando a ele acessar essas imagens por qualquer computador conectado à Internet.

Num modo complementar de operação, o usuário pode também telefonar para o sistema e, além de ouvir os dados das invasões registradas, pode selecionar se deseja ou não ser avisado da ocorrência das mesmas. Outro comando possível é o de apagar os dados das invasões anteriormente registradas.

O enfoque usado no desenvolvimento deste projeto é o de Sistemas a Eventos Discretos, que fornece as ferramentas adequadas para sua modelagem e implementação.

Nos dois capítulos seguintes, faremos uma breve incursão na política de software de código aberto e nos fundamentos do Asterisk. Em seguida, lançadas as bases para a compreensão da terminologia básica empregada, iniciaremos uma descrição minuciosa do sistema implementado.

---

[1] Para uma explanação da expressão "PBX digital", ver cap. 4.
[2] Ver cap. 4 e Apêndice A.
[3] O software de detecção de movimento é o que acompanha a WebCam, e será citado na seção 5.1.2.

3. O Linux e a Política de Código Aberto

Como foi citado no cap. 1, a implementação da interface computacional com a telefonia foi feita utilizando software de código aberto.

O servidor do PBX instalado é um PC operando com o Linux, que é o sistema operacional mais popular de código aberto. Para se ter uma idéia da abrangência deste sistema, citemos um levantamento [2] realizado em 2003 pela TechLab (divisão de pesquisas da consultoria E-Consulting) em 238 empresas brasileiras, o qual revelou que 78% delas utilizam o Linux como servidor em pelo menos uma aplicação.

Outro dado relevante é que, segundo a IDC [6], as plataformas de software livre estiveram entre as prioridades de investimento das médias empresas em 2005. Cerca de 13% das empresas apontaram como prioridade de investimento o desenvolvimento em soluções de plataforma aberta. Do total de médias empresas, 62% delas afirmaram investir parte do seu orçamento em software livre.

O Linux, com 14 anos de história (sua primeira versão oficial, a 0.02, foi anunciada em 5 de outubro de 1991), vem atingindo crescente popularidade não só para aplicações de médio e grande porte, mas também para o usuário doméstico. O advento das interfaces gráficas para este sistema transformou sua operação, antes baseada em linhas de comando e só acessível aos usuários iniciados, em algo tão simples quanto a do popular Windows.

É fato que existem alguns entraves iniciais, sobretudo para quem necessita utilizar elementos mais técnicos, mas, para o usuário habituado a iniciar um programa clicando em ícones, as diversas interfaces gráficas do Linux são bastante intuitivas.

A escolha deste sistema baseado na política de código aberto – aquela em que programadores do mundo todo têm acesso aos arquivos-fonte e podem realizar incrementos e novas construções sobre o sistema – tem diversas motivações:

- A já citada abrangência do Linux pelas empresas em seus servidores assegura que os sistemas nele baseados sejam receptíveis no mercado;
- O custo da instalação e da manutenção de um sistema Linux resume-se ao dos profissionais envolvidos nessas tarefas, uma vez que não há licenças como no caso dos sistemas proprietários;

- Trata-se de um sistema poderoso e confiável, características próprias dos softwares feitos com a política de código aberto;
- É de fácil obtenção;
- Há uma infinidade de ferramentas disponíveis na Internet, igualmente livres de licenças, que podem ser instaladas para dar suporte aos sistemas desenvolvidos (como o servidor de FTP utilizado neste projeto).

A partir desses elementos, justifica-se a preferência a um sistema baseado em código aberto em relação a um comercial, desde que haja ferramentas disponíveis para os fins da aplicação proposta.

Para o caso de uma aplicação envolvendo interface com telefonia e menus de voz (*Interactive Voice Response*), existe o Asterisk, uma potente ferramenta de código aberto feita para funcionar em Linux.

No Design-Lab, criou-se um ambiente de teste para este software em um PC rodando o Kurumin Linux[4]. Já o Asterisk, que ainda não está disponível em Português, foi instalado a partir dos pacotes da seção *stable* do Debian.

O capítulo seguinte ilustra as principais características do Asterisk e sua relação com a telefonia.

[4] O Kurumin Linux é uma distribuição do Linux feita por um brasileiro, e inclui uma série de aplicativos em português. Para maiores detalhes sobre esta e outras distribuições, consultar a referência [2].

4. O Asterisk e a Tecnologia de Voz Sobre IP

4.1. Indrodução

O Asterisk [7] é um software que possibilita a implementação de um PBX digital, com transmissão de dados em Voz Sobre IP (VoIP), e que pode ter conexão com a telefonia comum. Além disso, dispõe de uma linguagem para a construção de sistemas IVR[5] e de uma API[6] para interface com aplicativos externos.

Faz-se necessário explicar o uso do termo PBX, normalmente empregado para descrever as antigas centrais telefônicas, com ramais e operador(es). Com o surgimento da tecnologia de Voz Sobre IP, que possibilita a automação e a flexibilização dos sistemas envolvendo central telefônica e terminais de atendimento (operados por atendentes), acabou-se por utilizar parte da terminologia tradicional para nomear também os novos sistemas. Assim, quando usamos a expressão "PBX digital", estamos, na verdade, fazendo referência aos sistemas que fazem o intercâmbio entre ligações (originárias da linha telefônica comum ou de outros meios ditos "digitais", como a própria Voz sobre IP) e os terminais (computadores) aos quais estão os atendentes. Estes terminais, que são o equivalente digital dos antigos "ramais", são chamados, no jargão do Asterisk, de *extensões*, e o fluxo de dados de conversação entre eles e o servidor se dá através de VoIP.

[5] Sigla para Interactive Voice Response – Trata-se dos sistemas de auto-atendimento telefônico que oferecem funções ao usuário através de menus interativos baseados em mensagens pré-gravadas.
[6] Application Program Interface – conjunto de instruções (biblioteca) que dá acesso às funcionalidades de um sistema para algum outro programa.

4.2. A Telefonia Analógica

Uma das capacidades mais interessantes do Asterisk é justamente realizar a interface entre a telefonia analógica e o meio digital. Através do hardware específico, podemos conectar a linha telefônica usual ao computador, e então o Asterisk estará apto a receber e enviar dados através dessa linha, seja para efetuar uma discagem, seja para conectá-la a uma extensão em VoIP, seja para atender automaticamente uma ligação e emitir mensagens de voz pré-gravadas para o usuário, podendo inclusive interagir com o mesmo por meio do reconhecimento dos sinais DTMF[7] por ele digitados.

Denomina-se PSTN (*Public Switched Telephone Network*) a rede de telefonia pública tradicional, que tem por objetivo tão somente estabelecer e manter uma conexão de áudio entre dois pontos determinados.

Atualmente, no Brasil, a parte analógica da linha telefônica está apenas na seção que liga as tomadas telefônicas das residências aos conversores instalados pela operadora em cada setor (prédio ou grupo de casas). A partir daí, faz-se a amostragem / digitalização dos dados de áudio e sua transmissão ocorre por meios digitais[8] (como fibras óticas).

Entretanto, a conexão que se encontra nas tomadas telefônicas das residências (ou estabelecimentos em geral) é analógica.

A fim de permitir que o Asterisk receba / transmita dados pela linha telefônica, utilizamos uma placa com duas tomadas idênticas à dos telefones; uma delas liga-se à linha telefônica e a outra ao aparelho telefônico comum. Assim, pode-se continuar usando o aparelho telefônico normalmente, e tem-se a possibilidade de direcionar (manual ou automaticamente) o atendimento da linha para o Asterisk.

---

[7] Sigla para Dual-Tone Multi Freqüency, que é o sistema atualmente usado para a discagem nos telefones comuns.

[8] Uma descrição abrangente das formas de digitalização utilizadas na PSTN pode ser encontrada em [3].

4.3. Voz Sobre IP

O Asterisk não se limita à interface com a telefonia tradicional. Possui também a capacidade de gerenciar fluxos de voz através das conexões de rede, usando a tecnologia de Voz Sobre IP (ou VoIP). Tal capacidade foi também explorada neste projeto.

4.3.1. Mercado

A tecnologia Voz sobre IP está entre as prioridades de investimentos do mercado SMB (small and medium Business) brasileiro, segundo a IDC [6]. As médias empresas já são foco de investimentos dos grandes fornecedores de tecnologia pelo potencial de mercado que representam não só no Brasil, mas em nível mundial.

A IDC verificou que os assuntos de maior interesse para os gestores da área de tecnologia das médias empresas são segurança, infra-estrutura, VoIP, ERP (Planejamento de Recurso Empresarial) e Business Intelligence.

VoIP, infra-estrutura e ERP são assuntos ligados ao ganho de produtividade do negócio da empresa. Dentre os benefícios das empresas na adoção de VoIP estão a melhoria da comunicação, mobilidade e redução de custos.

4.3.2. Funcionamento

A premissa básica [3] da tecnologia de Voz Sobre IP é o empacotamento de fluxos de áudio para o transporte através de redes baseadas em IP (*Internet Protocol*). O grande desafio para se realizar esta tarefa reside na maneira como as pessoas se comunicam. Não só o sinal precisa chegar essencialmente na mesma forma em que foi transmitido, como também precisa fazê-lo em menos de 300 milissegundos. Se os pacotes forem perdidos ou atrasados, haverá degradação na qualidade da experiência da comunicação.

Os protocolos de transporte, que são coletivamente chamados "a Internet", não foram originalmente projetados para transmissão de mídia em tempo real. Os pontos de destino tinham sido concebidos para resolver o problema de pacotes perdidos esperando um pouco mais até que eles chegassem, solicitando retransmissão ou, em alguns casos, considerando a

informação perdida para sempre e simplesmente continuando sem ela. Para transmitir uma típica conversação de voz, esses mecanismos não servirão. Nossas conversas não se adaptam bem à perda de letras ou palavras, nem fazem algum atraso considerável entre a transmissão e o recebimento.

A PSTN (*Public Switched Telephone Network*) tradicional foi projetada especificamente com o propósito de transmissão de voz, e serve perfeitamente a este objetivo do ponto de vista técnico. Já do ponto de vista da flexibilidade, entretanto, suas imperfeições são evidentes. A VoIP tem a promessa de incorporar comunicações de voz entre todos os outros protocolos que carregamos em nossas redes, mas, devido às demandas especiais de uma conversação de voz, técnicas especiais são requeridas para projetar, construir e manter essas redes.

O problema envolvendo a transmissão de voz baseada em pacotes vem do fato de que a forma com que nós falamos é totalmente incompatível com a forma em que o IP transmite dados. Falar e ouvir é um constante revezamento de fluxos de áudio, enquanto que os protocolos da Internet são feitos para cortar tudo em pedaços, encapsular os bits de informação em milhares de pacotes, e então entregar cada pacote de qualquer jeito possível à outra extremidade. Evidentemente, algum tipo de ponte é necessário.

4.4. Protocolos de VoIP – o SIP

O mecanismo para carregar uma conexão de VoIP geralmente envolve uma série de transações de sinalização entre as extremidades da linha de comunicação (no meio da qual estão os portais), culminando em dois fluxos de mídia persistentes (um para cada direção) que carregam a conversação real. Existem diversos protocolos para manipular isso, entre eles o IAX, o H.323 e o SIP. O Asterisk suporta todos eles, mas para este trabalho foi utilizado o SIP (*Session Initiation Protocol)*.

O SIP vem sendo cada vez mais utilizado no mundo da VoIP. Sua premissa é que cada extremidade de uma conexão é um porto (*peer*), e o protocolo negocia capacidades entre eles. O que torna o SIP instigante é que ele é um protocolo relativamente simples, com uma sintaxe semelhante àquela de outros protocolos familiares como o HTTP e o

SMTP. Além disso, suas possibilidades estão se expandindo para além da VoIP. Num futuro próximo, deverá incluir a habilidade de transmitir vídeo, música, e qualquer tipo de multimídia em tempo real. O SIP está fadado carregar a maioria das novas aplicações a partir dos próximos anos.

4.5. Fontes de Informação

Face à velocidade com que as tecnologias de comunicação têm avançado, são poucos os artigos técnicos formais que abordam o uso de aplicações como o Asterisk. Muitas informações preciosas são hoje obtidas através da Internet, em contextos pouco formais, como sites de software livre e fóruns de discussão.

É freqüente, porém, a participação de pesquisadores de importantes universidades nessa profusão "*off-science*" de informações tecnológicas que se dá pela rede global.

No Apêndice A, há uma introdução sobre a configuração básica do Asterisk e sobre o desenvolvimento de *DialPlans*. Alguns websites também foram listados na bibliografia, onde informações adicionais podem ser encontradas.

5. Detalhes da Implementação

5.1. Componentes

Mostraremos aqui os componentes do sistema desenvolvido, bem como as inter-relações entre os mesmos.

5.1.1. Hardware

Os seguintes elementos foram utilizados na concepção do sistema:

- 3 PC's, dois deles rodando os clientes SIP (ramais) e outro rodando o Asterisk (o Asterisk Server);
- 1 placa da Digium, modelo X100-P, conectada no Asterisk Server, para a interface com a entrada analógica da linha telefônica;
- 1 aparelho de telefone comum, ligado na placa X100-P. Trata-se do telefone do Laboratório, que opera normalmente quando o Asterisk não está ativado;
- 1 webcam comum, da marca Creative Labs, acoplada a um dos computadores que rodam os ramais (no caso, o 2000).

A figura seguinte ilustra os componentes e suas interconexões:

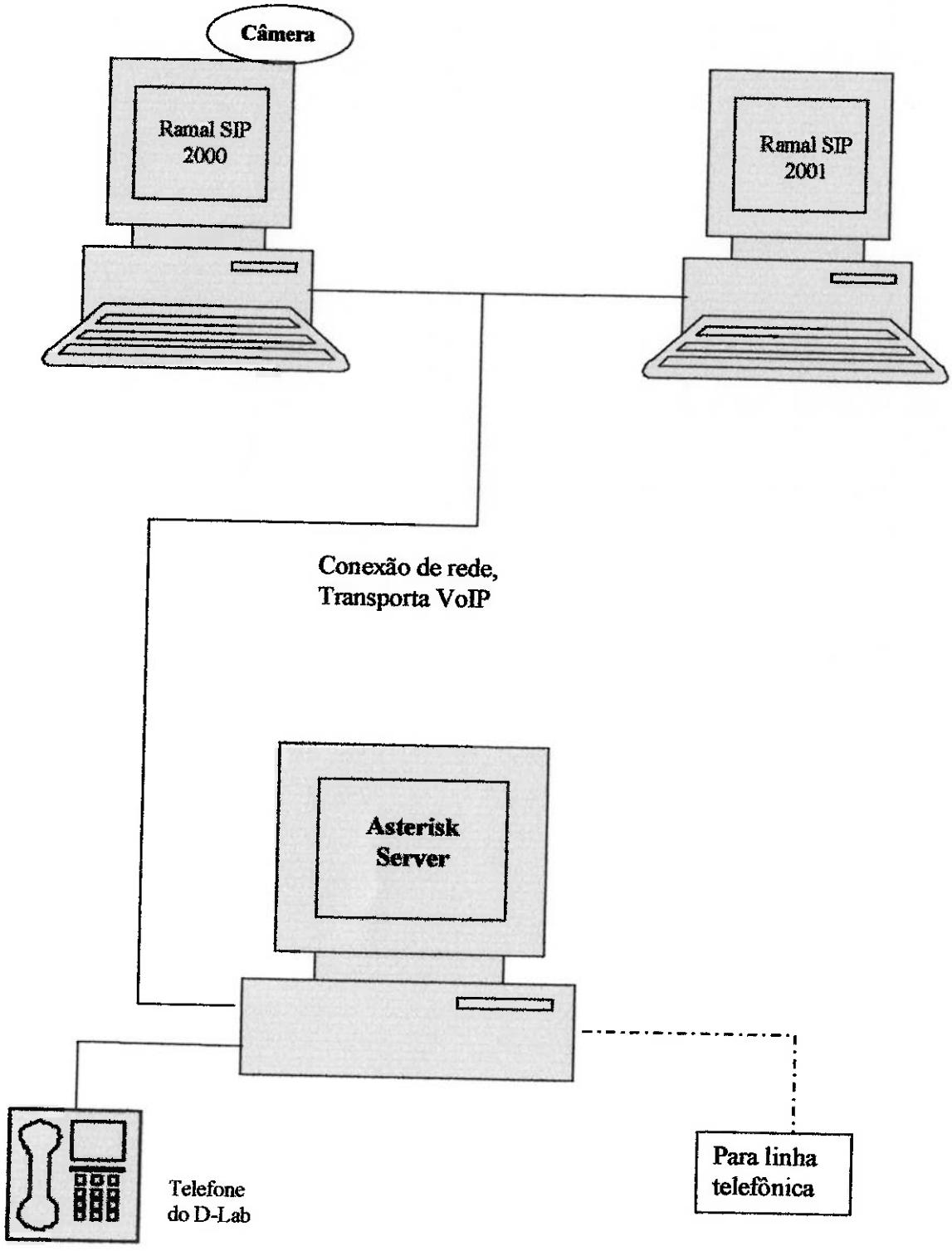


Figura 5.1 - Esquema das conexões físicas entre os componentes

5.1.2. Software

Os seguintes softwares foram instalados a fim de realizar a operação do sistema:

- No Asterisk Server (sistema operacional Linux): Asterisk v.1.0.7; Zaptel v.1.2 [12]; gFTPd (servidor de FTP) [14]. Nesta máquina foi também colocado um script que roda de 1 em 1 minuto para verificar se foi recebido o arquivo que informa a ocorrência da detecção de movimento (ver Apêndice B).
- Em ambos os ramais SIP (sistema operacional Windows XP): cliente SIP XTEN XLite (softphone) [16]. A configuração do XLite é feita colocando-se em "Authorization User" e em "Password" os dados correspondentes que estão no arquivo `/etc/asterisk/sip.conf`. O endereço IP do Asterisk Server também precisa ser indicado na seção "domain/realm", e, em "SIP Proxy", coloca-se o mesmo entereço IP seguido de ":5060", que é a porta configurada para o SIP no Asterisk em `/etc/asterisk/sip.conf` (ver Apêndice B).
- No PC do ramal SIP 2000: Creative WebCam Center v.1.00.09.

O Zaptel é um módulo necessário para o funcionamento da placa X100-P (ou qualquer outra que faz interface com a linha telefônica), e sua instalação é feita conforme explicado no Apêndice A.

5.2. Modelo do Sistema

Mostraremos, abaixo, os modelos em Redes de Petri para o dispositivo proposto.

As Redes de Petri possibilitam visualizar com facilidade a operação do sistema que foi dividida, para facilitar a implementação, em 2 formas de funcionamento: a solicitada e a automática.

Observe que toda a interação entre o usuário e o controlador se dá por telefone, através de um sistema de IVR (Interactive Voice Response), designado de acordo com as necessidades do projeto.

5.2.1. Operação em Modo Solicitado

Esta forma de operação consiste na situação em que o usuário liga para a central com o intuito de ativar ou desativar o modo de supervisão automática. É fato que nem sempre será do interesse do usuário receber as ligações avisando da detecção de movimento. Assim, ele terá a opção de ativar ou desativar o monitoramento através de uma ligação para o controlador.

Para este caso, a seqüência de atividades do sistema será como mostra a figura a seguir.

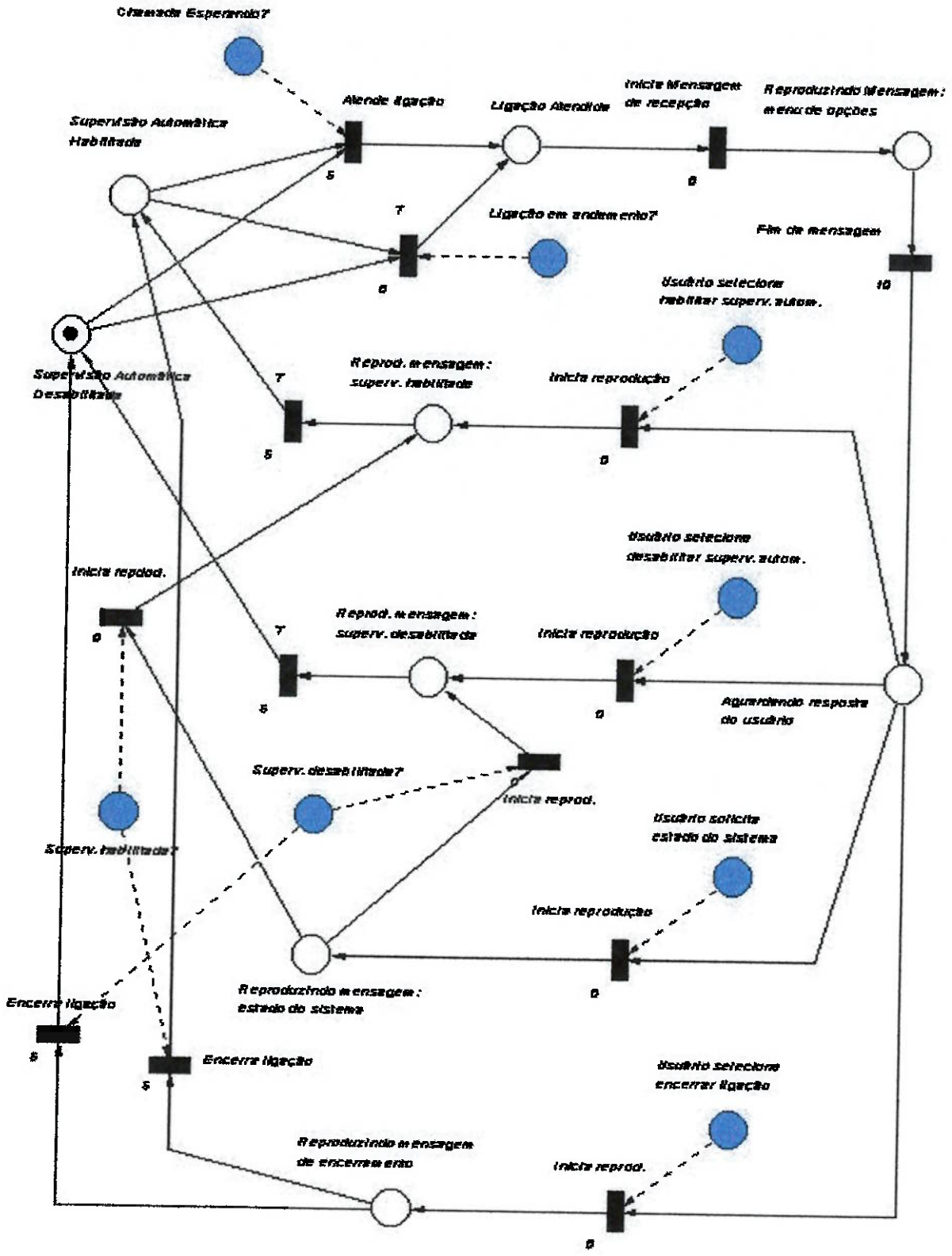


Figura 5.2 - Modelo da Operação em Modo Solicitado

5.2.2. Operação em Modo Automático

Aqui temos a operação principal do sistema: a supervisão automática do ambiente.

A partir das imagens capturadas pela webcam, instalada no PC que roda o ramal SIP 2000, o software da mesma detectará se houve mudança significativa em relação ao histórico de imagens, configurando intrusão. Se isso ocorrer, este software enviará um sinal por FTP ao servidor do Asterisk. Um script programado no servidor roda de 1 em 1 minuto verificando a chegada do sinal. Caso isso ocorra, aciona o Asterisk, que telefona para o usuário, avisando da detecção de intrusão. O aplicativo também envia o primeiro quadro capturado para o e-mail do usuário (imagem JPG).

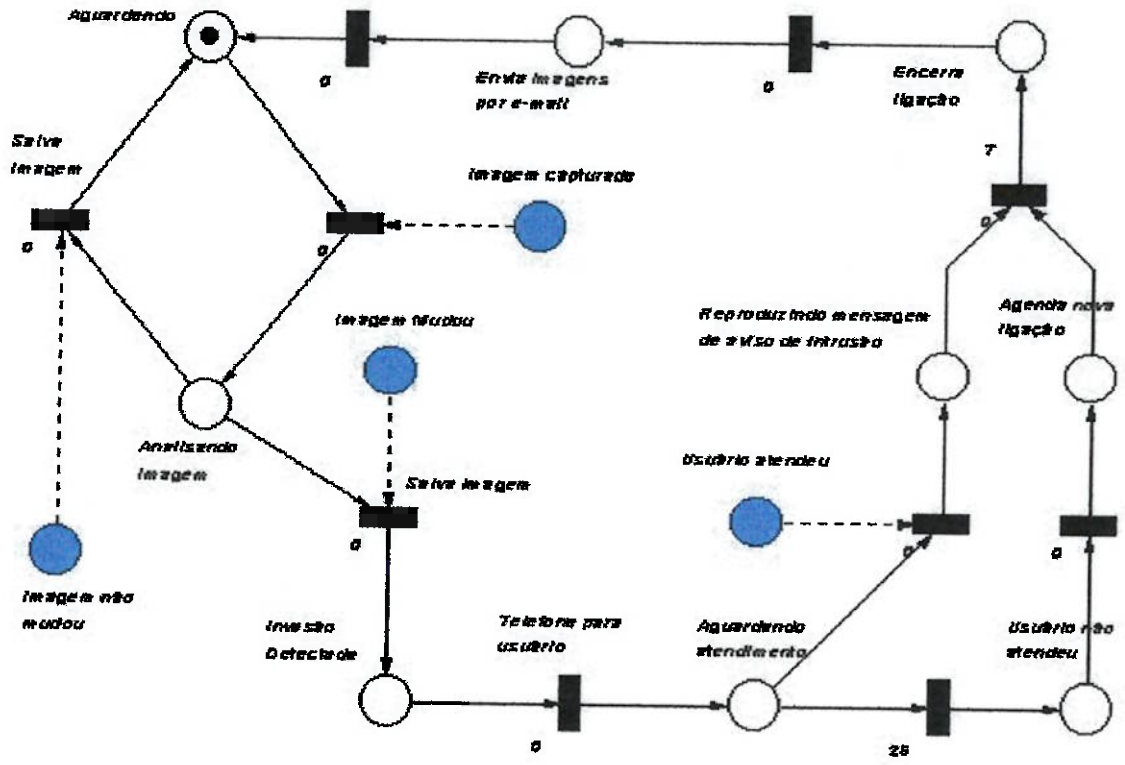


Figura 5.3 - Modelo da Operação em Modo Automático

5.3. O PBX Instalado

O sistema de IVR (menu de voz) do monitoramento automático, que foi chamado de D-Monitor, foi implementado como um dos ramais do PBX do Desing-Lab. Assim, no servidor do Asterisk, foi implementado um DialPlan[9] que recebe ligações de um usuário externo pela linha telefônica, e solicita a ele o número do ramal desejado (por meio de mensagens pré-gravadas).

Se o usuário digitar 2000 ou 2001, o DialPlan direciona-o para o ramal SIP correspondente. Assim, caso o operador correspondente esteja com o XLite (softphone SIP) ativado, seu telefone virtual irá tocar, e ele poderá atender a ligação fazendo uso de um headphone com microfone embutido para comunicar-se com a outra parte.

Por outro lado, se o usuário digitar 3111, que é o ramal escolhido para o D-Monitor, o Asterisk direcionará a ligação para o contexto do D-Monitor, que funciona exclusivamente por meio da reprodução de mensagens pré-gravadas para o usuário, e da leitura dos dígitos teclados pelo mesmo. O menu é auto-explicativo, e sua dinâmica é explicitada no DialPlan listado no Apêndice B.

[9] Literalmente, "plano de discagem" – trata-se da essência de qualquer sistema Asterisk, que é o arquivo que define como o Asterisk recebe cada ligação e como ele a direciona. Este arquivo é chamado *extensions.conf*.

6. Aplicações em Automação Predial

Nos dias atuais, é constante a presença de sistemas de IVR (*Interactive Voice Response*) nas empresas dos mais diversos segmentos: bancos, escritórios, lojas virtuais, cinemas, teatros etc.

O objetivo desses sistemas não é reduzir o contingente de profissionais necessário para o atendimento dos clientes, tendo em vista que, no quesito de relação com o cliente, não há máquina que substitua o ser humano. Essencialmente, usam-se os dispositivos automáticos para efetuar tarefas cuja realização por pessoas seria impossível ou inviável. Por exemplo, não se pode confiar a pessoas o trabalho de informar o extrato bancário de clientes por telefone. Além da susceptibilidade a erros, simplesmente não há como criar, para esse fim, uma estrutura de atendimento telefônico que dê conta de altas demandas baseando-se exclusivamente no atendimento pessoa-a-pessoa. Assim, direcionam-se as ligações dos clientes para árvores de voz automáticas, que realizam inúmeras funções, e filtram para o atendente humano apenas os casos em que este atendimento realmente é necessário.

Já nos casos de assistência técnica por telefone, em que o atendimento por um profissional é sempre necessário, os sistemas de IVR são usados para distribuir eqüitativamente as chamadas entrantes para os diversos atendentes, e/ou para direcionar cada uma delas a um profissional capacitado para respondê-la.

Os sistemas baseados em Asterisk prometem subverter a noção clássica que se tem sobre a finalidade dos sistemas de IVR.

O Asterisk proporciona funcionalidades que estão muito além daquelas necessárias para a concepção dos *Call Centers* tradicionais.

Em primeiro lugar, os IVR's feitos com o Asterisk têm um grau de flexibilidade que é impensável para os sistemas comerciais que usam tecnologia proprietária. Tais sistemas são construídos com uma pesada estrutura de software que não raramente requer a instalação de caros servidores e dispositivos de hardware para rodá-la. Já o Asterisk tem uma adaptabilidade admirável à estrutura pré-existente. Requer, sim, um hardware específico,

mas sua estrutura aberta permite uma intimidade muito maior com os sistemas projetados, facilitando alterações ou acréscimo de novas funções.

Por exemplo: um sistema comercial que não foi concebido para dar suporte a SIP para chamadas externas necessitará de um alto investimento para fazê-lo. Talvez precise ser totalmente redesenhado. Já o Asterisk tem em sua concepção o suporte aos principais protocolos de Voz Sobre IP, e, com o simples acréscimo de algumas linhas de comando no DialPlan, estará configurado para suportar chamadas externas via SIP.

Outro grande diferencial entre o Asterisk e os sistemas de IVR com tecnologia proprietária é sua capacidade única de interagir com programas externos. O Asterisk dispõe de uma interface padrão chamada AGI – *Asterisk Gateway Interface* – que permite, através de scripts, que o DialPlan seja controlado por aplicativos externos, além de poder fornecer dados e comandos para os mesmos (comunicando-se, por exemplo, com um banco de dados externo)[10].

É fácil perceber que esta capacidade abre possibilidades infinitas para os sistemas de telefonia. Com a habilidade de comunicar-se com aplicativos externos, faz-se possível ao usuário acessar qualquer função controlável por computador através de seu telefone, desde um simples monitoramento até o controle ativo de aparelhos.

Por exemplo, uma pessoa que está fora de sua cidade, ou de seu país, poderá ativar ou desativar o alarme de sua casa com um simples telefonema. Poderá também ser avisada pelo sistema quando o alarme disparar, recebendo uma ligação telefônica automática, em que a gravação pode inclusive informar qual sensor disparou o alarme. O aviso remoto de alarme disparado servirá, sobretudo, para donos de estabelecimentos comerciais cujos alarmes freqüentemente disparam sem qualquer razão aparente, durante a madrugada.

Ademais, não nos esqueçamos de que o SIP permitirá unir, num processo de comunicação em tempo real, todo tipo de mídia (voz, imagens, músicas, etc).

[10] O Asterisk possui um banco de dados interno, que foi utilizado neste projeto, e que é acessível através de comandos específicos no dialplan (ver Apêndice A).

7. Conclusão

Este trabalho reside na vanguarda de um setor tecnológico cujo avanço tende a tomar rumos fascinantes nos próximos anos: a comunicação remota homem-máquina por meio de mensagens audíveis em tempo real.

Foram exploradas diversas funcionalidades da mais nova e promissora ferramenta computacional de gerenciamento de conexões de voz, seja pela linha telefônica, seja pela mídia digital (rede ou Internet).

Abordamos o protocolo SIP, que está se tornando dominante na sinalização e transporte de dados instantaneamente, com aplicações fundamentais dentro da tecnologia de Voz Sobre IP.

O Asterisk tem uma imensa gama de aplicações, desde a implementação de PBX's locais até o gerenciamento de conexões VoIP em múltiplos protocolos em plataformas de grande porte com ligações para qualquer parte do mundo, incluindo funções para transmissão de imagens e outros dados.

Seu uso, no presente trabalho, deu-se através da construção de um sistema de IVR que recebe dados de uma aplicação externa, e interage com o usuário via ligação telefônica. Além disso, utilizamos as funções de interface de dados de Voz Sobre IP com a linha telefônica ordinária.

O objetivo foi demonstrar uma das possibilidades de uso do Asterisk, além de abrir uma porta e fornecer as bases para o desenvolvimento de novas aplicações, que façam uso de outras características dessa poderosa ferramenta, cujo potencial já é, literalmente, ilimitado.

8. Bibliografia

8.1. Livros e Artigos

Obs.: Todos os sites listados foram acessados em 14 de março de 2006.

1. Murata, T. – "Petri Nets: Properties, Analysis and Applications", 1989, Proceedings of IEEE, vol. 77, no. 4.

2. Morimoto, C. – "Entendendo e Dominando o Linux", 2004, Digerati Books, São Paulo.

3. Meggelen, J. V.; Smith, J.; Madsen, L. – "Asterisk: The Future of Telephony", 2005, O'Reilly Media, Inc., disponível em: < http://www.asteriskdocs.org >.

4. Todd, J. – "Asterisk: A Bare-Bones VoIP Example", 2003, ONLAMP Publications, disponível em: < http://www.onlamp.com/pub/a/onlamp/2003/07/03/asterisk.html >

5. Todd, J. – "VoIP and POTS Integration with Asterisk", 2004, ONLAMP Publications, disponível em:
< http://www.onlamp.com/pub/a/onlamp/2004/01/22/asterisk2.html >

6. Gaia, D. – "Brazil IT Spending Trends Midmarket", 2005, IDC.

8.2. Sites da Web

7. Asterisk – Site Oficial: < http://www.arterisk.org >

8. Documentação do Asterisk: < http://www.asteriskdocs.org >

9. Informações sobre VoIP, além de um guia extensivo sobre o Asterisk e suas funções, desde as básicas até as avançadas: < http://www.voip-info.org >

10. Fórum brasileiro de discussão sobre o Asterisk: < http://www.asteriskbrasil.org >

11. Instalação do Zaptel e da placa X100-P:
    a. < http://www.voip-info.org/tiki-index.php?page=Asterisk%20Zaptel%20Installation >
    b. < http://ftp.digium.com/pub/asterisk/INSTALL.README.X100P >

12. Download do Asterisk e programas relacionados: < ftp://asteriskpbx.com >

13. Fórum de discussão do Kurumin Linux: < http://www.kuruminlinux.com.br >

14. Download do gFTPd: < http://www.gftpd.org/ >

15. Guia sobre scripts no Linux:
    < http://howtos.linux.com/guides/abs-guide/index.shtml >

16. X-Ten's X-Lite (softphone SIP): < http://www.xten.com/ >

17. Site da Digium (hardware para interface do Asterisk com a telefonia):
    < http://www.digium.com >

Apêndice A – Uma Breve Introdução ao Asterisk

1. Instalação

Há duas formas de se instalar o Asterisk.

A primeira é através do *apt-get* do Linux, que no Kurumin pode ser feita pelos Ícones Mágicos. Basta digitar 'asterisk' no campo do nome do pacote a ser instalado, na janela *Instalar Novos Programas*, e escolher 'Instalar'. Com isso, o Asterisk será instalado sem os arquivos-fonte, ou seja, somente a versão compilada, pronta para ser utilizada.

A segunda forma consiste em baixar o pacote com os fontes e compilá-los no computador. Não cobriremos esta forma aqui, mas ela está detalhadamente descrita na referência [3]. Em vez de baixar o pacote a partir do CVS, pode-se baixá-lo a partir de [12] (ver bibliografia).

Importante: para as aplicações que utilizarão o hardware da Digium (ou equivalente) para fazer interface com a linha telefônica, faz-se necessária a instalação do ZapTel, que é o módulo que controla a placa através de drivers. A única forma que funcionou, nas tentativas que fizemos, foi através da explicação em [11-a], ou seja, baixando os fontes e compilando-os na máquina. A diferença com relação ao explicado no site é que os fontes não foram baixados do CVS, mas a partir do FTP de [12]. Outro local a se obterem os fontes do Asterisk e do Zaptel é o FTP indicado em [11-b].

2. Configuração

A configuração inicial do Asterisk foi feita baseando-se nos exemplos fornecidos em [5]. Basta copiar os arquivos 'conf' sobre os originais (após efetuar backup destes) e deixar todos os outros como estão.

A partir daí, pode-se configurar os DialPlan através da edição dos seguintes arquivos:

/etc/asterisk/extensions.conf – Define o DialPlan, ou seja, a operação do Asterisk frente às ligações telefônicas.

/etc/asterisk/sip.conf – Define os ramais SIP do PBX a ser configurado, e também os dados de chamadas SIP externas. Define os contextos de *extensions.conf* que recebem cada tipo de chamada SIP.

/etc/asterisk/voicemail.conf – Define dados da caixa-postal (correio de voz). Além de editar este arquivo, deve-se rodar o aplicativo 'addmailbox', especificando os dados das caixas postais a serem criadas [4].

/etc/asterisk/zapata.conf – Define as opções de sinalização da placa, bem como o contexto de *extensions.conf* para onde são direcionadas as chamadas provenientes da linha telefônica.

3. Desenvolvimento do DialPlan

Uma vez realizadas as partes "burocráticas" (instalação dos programas e configuração do hardware e da estrutura física do PBX), podemos partir para a programação do DialPlan, que se dá exclusivamente no arquivo **/etc/asterisk/extensions.conf**.

A referência [3], que pode ser baixada diretamente de <http://www.asteriskdocs.org/modules/tinycontent/index.php?id=11>, apresenta uma explicação detalhada sobre o design do DialPlan.

No Apêndice B estão listados todos os arquivos editados para a realização deste trabalho, incluindo o *extensions.conf*. Alguns aspectos que merecem ser aqui citados, que serão melhor compreendidos a partir da leitura deste arquivo, são:

- O DialPlan é dividido em *contextos*. Os contextos são seções de instruções com uma determinada finalidade. Por exemplo, o contexto [globals] tem instruções que definem variáveis globais, cujos valores são acessíveis por todas as instruções do DialPlan. Outra finalidade dos contextos é organizar os diversos segmentos da operação do DialPlan, separando, por exemplo, as partes referentes ao atendimento de uma ligação externa daquelas referentes à manipulação de uma chamada SIP. Os contextos são iniciados com um nome (arbitrário, com exceção do *globals* e do *general*) entre colchetes.

- *Extensões*. Este é um conceito fundamental no Asterisk. Extensões são entidades que referenciam conexões de voz. Toda e qualquer instrução que realiza alguma atividade referente ao atendimento, direcionamento e realização de operações lógicas sobre uma ligação, está necessariamente associada a uma extensão. Uma extensão pode ser um número (nos casos em que representam um ramal ou telefone específico), uma máscara (trechos de texto que podem representar um número genérico) ou determinadas letras, como a s (extensão padrão), i (extensão inválida), t (que é chamada quando ocorre um *timeout*) e h (de hangup – é chamada quando ocorre o desligamento da outra parte, nos casos em que o Asterisk está respondendo a uma chamada). Todos esses tipos de extensões foram explorados neste trabalho.

- *Prioridades*. Uma prioridade corresponde a uma linha de instrução, ou, mais especificamente, ao número que a representa. Em um DialPlan, qualquer instrução de manipulação de chamadas está associado a uma extensão e a uma prioridade. As prioridades numeram as instruções na exata ordem em que elas devem ser executadas. Instruções consecutivas devem ter prioridades consecutivas.

- *Canal*. Um canal é uma via pela qual uma ligação pode ser feita. Neste trabalho, definimos dois canais: o SIP (para os ramais externos) e o ZAP/1 (para acesso à linha telefônica).

- *Aplicativos*. O Asterisk está munido de dezenas de aplicativos para realizar as mais variadas funções. Desde atender automaticamente uma ligação após esperar o telefone normal chamar por 20 segundos, até reproduzir uma mensagem pré-gravada, ou ligar para determinado número telefônico, são operações realizadas através de aplicativos.

- *Variáveis*. O uso de variáveis confere grande poder aos dialplans. Pode-se, através delas, contar quantas vezes o usuário digitou a senha errada, para desligar após o terceiro erro. Pode-se também guardar o nome ou o número de um usuário remoto para, oportunamente, chamá-lo. A manipulação de variáveis é muito flexível, permitindo inclusive operações lógicas e matemáticas, e manipulação de strings. Existem as variáveis globais e as variáveis de canal. As globais são definidas na seção [globals] no início do dialplan, ou criadas chamando-se a aplicação `SetGlobalVar()`, e são acessíveis em qualquer ponto do dialplan. Já as variáveis de canais, que são criadas

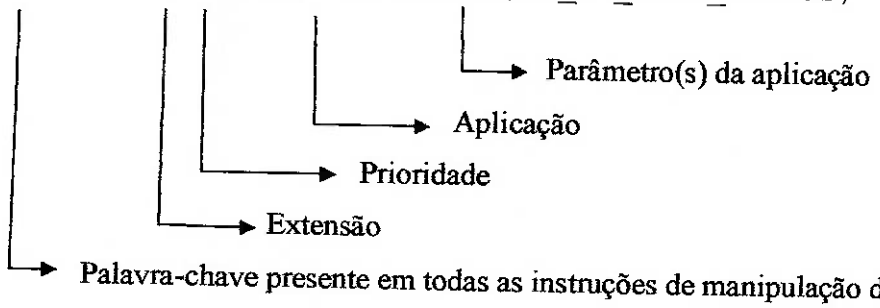
através da aplicação `SetVar()`, são acessíveis somente no canal atualmente em uso. Assim, se uma variável de canal for criada durante o atendimento de uma ligação do canal ZAP/1, e se a ligação for transferida para um canal SIP (como o ramal 2001), o valor da variável se perderá. Normalmente (mas não obrigatoriamente) usam-se nomes de variáveis em letras maiúsculas. O valor de uma variável é obtido através do indicador ${}. Assim, o valor de uma variável chamada COUNTER será expresso por ${COUNTER}.

- *Operações lógicas e Redirecionamento.* Nem sempre a operação de um dialplan é linear. Freqüentemente será necessário redirecionar o fluxo da operação do dialplan para alguma prioridade que não é a consecutiva à atual. Por exemplo, durante o dia queremos que nosso sistema atenda com a mensagem de "bom dia", durante a tarde, queremos um "boa tarde", e um "boa noite" para a noite. Isso não pode ser feito sem algum tipo de lógica condicional. O Asterisk fornece, portanto, meios de se redirecionar o fluxo do programa para uma prioridade não-consecutiva, que nem precisa estar no mesmo contexto atualmente em execução. Para isso, usam-se as aplicações `Goto()`, `GotoIf()` e `GotoIfTime()`. No dialplan do D-Monitor, todas são utilizadas.

- *Expressões lógicas e operações matemáticas.* Os dialplans do Asterisk são dotados de capacidades lógicas semelhantes às linguagens de programação. Aqui, vale a pena mencionar que as operações lógicas e matemáticas precisam estar sob o indicador $[]. Deste modo, para somarmos 1 à variável de canal CONT e armazenar o valor na própria CONT, escreveríamos: `SetVar(CONT=$[${CONT} + 1])`.

Um exemplo de uma típica instrução de um dialplan seria:

```
exten => s,2,Playback(mensagem_de_boas_vindas)
```



Apêndice B – Listagem dos Arquivos – Fonte

/etc/asterisk/zapata.conf

```
;
; Zapata telephony interface
;
; Arquivo de configuração
;
; ------------------------------------------------------------
; /etc/asterisk/zapata.conf
;
; por José Adriano M. Pardi, novembro de 2005.
; ------------------------------------------------------------
;
; A versão original do Asterisk tem um exemplo bem
; completo deste arquivo. Esta é uma versão mais simples,
; que é suficiente para os propósitos deste trabalho.
;
[channels]
language=en
context=from-analog  ; Este é o contexto de 'extension.conf' que receberá
                     ; as ligações oriundas da linha telefônica.
;
signalling=fxs_ks    ; Forma de sinalização da placa (simula um aparelho
                     ; telefônico (dispositivo FXS)
usecallerid=yes
echocancel=yes
echocancelwhenbridged=yes
channel => 1         ; Canal configurado no Zaptel.
```

/etc/asterisk/sip.conf

```
; ------------------------------------------------------------
; /etc/asterisk/sip.conf
;
; por José Adriano M. Pardi, novembro de 2005.
; ------------------------------------------------------------
;
[general]
port=5060                ; Porta à qual se conectar (SIP é 5060)
bindaddr=0.0.0.0         ; Endereço para se conectar (todos os da máquina)
allow=all
context=from-sip-external ; Envie chamadas SIP de origem desconhecida
para
                          ; este contexto.
;
; Ramal 2000:
;
[2000]
type=friend               ; "friend" significa que esse dispositivo faz e
                          ; recebe chamadas.
username=2000             ; Corresponde ao "Authorization user" no
softphone
secret=2000               ; Senha para este usuário
host=dynamic              ; Permite que este ramal não esteja sempre no
                          ; mesmo IP
context=from-sip-internal ; Chamadas entrantes deste ramal vão para este
                          ; contexto do 'extensions.conf'
mailbox=2000              ; Correio de voz correspondente a este ramal
                          ;
canreinvite=no            ; Obriga o fluxo a passar sempre pelo servidor
;
; Ramal 2001
;
[2001]                    ; Mesmo que o 2000, mas com dados de
autenticação
type=friend               ; (usuário e senha) diferentes.
username=2001
secret=2001
host=dynamic
context=from-sip-internal
mailbox=2001
canreinvite=no
```

/etc/asterisk/extensions.conf

```
; ------------------------------------------------------------------------
; /etc/asterisk/extensions.conf
; ------------------------------------------------------------------------
; Este é o arquivo que define a operação do Asterisk.
; Aqui colocamos a seqüência de ações a ser seguida em cada caso,
; incluindo as chamadas feitas pelos usuários dos ramais e as chamadas
; provenientes da linha analógica.
;
; Aqui também está definida a operação da interface com o usuário do
; D-Monitor, envolvendo mensagens automáticas e menu de voz.
;
; As diversas instruções estão separadas em seções denominadas contextos,
; que são explicados no apêndice "Breve Introdução ao Asterisk".
; ------------------------------------------------------------------------
; Autor: José Adriano Meireles Pardi
; Data: outubro de 2005 a fevereiro de 2006
; ------------------------------------------------------------------------
;
;
[globals]
;
; Esta seção define as variáveis globais para o sistema:
;
; Diretório dos arquivos de som:
SOUND_DIR=/usr/share/asterisk/sounds/DMonitor
;
; Tempo de chamada antes do Asterisk atender:
TEMPO_CHAMADA=12
;
; Tempo máximo de chamada para um ramal em particular:
TEMPO_RAMAL=20
;
; Tempo de espera até a chamada ao ramal padrão:
TEMPO_PADRAO=10
;
; Tempo máximo de duração das ligações que vão para caixa-postal:
```

```
TEMPO_MAX_LIGACAO=90
;
; Tempo máximo que o sistema aguarda a digitação de um dígito:
TEMPO_MAX_DIGITO=5
;
; Tempo máximo que o sistema aguarda a resposta do usuário
TEMPO_MAX_RESPOSTA=5
;
; Máxima duração das chamadas automáticas (em segundos):
MAX_DURACAO_CHAMADA_AUTO=300
;
; Ramal padrão:
RAMAL_PADRAO=2000
;
; Ramal do D-Monitor
RAMAL_DMONITOR=3111
;
; Número do usuário D-Monitor
NUM_USUARIO_DM=3197
;
;
;
[general]
;
; Define algumas propriedades da operação do Asterisk:
;
static=yes        ; Estas duas linhas impedem que a interface de linha de
writeprotect=yes ; comando cause alteração de dados neste arquivo conf.
;
;
;
[from-sip-external]
;
; Este contexto recebe as ligações SIP provenientes de chamadores
; desconhecidos (sem registro no arquivo 'sip.conf').
; O nome deste contexto está especificado no próprio 'sip.conf'.
;
; Vamos possibilitar que chamadores externos se conectem com
```

```
; os ramais do sistema, através da inclusão do contexto
; [local-extensions]
; (lembremos que o D-Monitor é um dos ramais internos).
;
; A linha abaixo nada mais faz do que "copiar" todo o conteúdo do
; referido contexto para cá, permitindo que suas extensões/prioridades
; tenham efeito para as ligações que aqui caem.
;
; Um DialPlan bem organizado está dividido em diversos contextos, e o
; conteúdo de alguns deles é utilizado através da instrução 'include'.
; Isso posibilita que o mesmo bloco de código possa ser utilizado em
; diferentes situações sem que seja preciso duplicá-lo, o que traria
; não só um aumento desnecesário do tamanho do arquivo, como também
; muito mais esforço com a redundância na hora de fazer alterações.
;
; Observe que o mesmo contexto [local-extensions] também será incluído
; no contexto abaixo, que manipula as chamadas oriundas de ramais
; internos.
;
include => local-extensions
;
; Abaixo, a extensão 'h' significa que a ligação foi encerrada
; pela outra parte. Devemos colocar essa extensão em todos os
; contextos que manipulam chamadas diretamente.
;
; A prioridade abaixo possibilita que o Asterisk faça o devido
; encerramento de operação no caso de desligamento da outra parte.
;
exten => h,1,Hangup
;
; Se o usuário discou um número inválido (não previsto anteriormente),
; a ligação será direcionada para a extensão 'i' (de 'invalid').
; Aqui, mudamos a duração máxima da ligação para 15 segundos, tocamos
; um sinal de linha congestionada e, em seguida, desligamos.
;
exten => i,1,AbsoluteTimeout(15)
exten => i,2,Congestion
exten => i,3,Hangup
```

```
;
;
;
[from-sip-internal]
;
; Chamadas originárias de algum ramal SIP interno do nosso sistema
; serão tratadas aqui (veja o arquivo 'sip.conf').
;
; Da mesma forma como acima, vamos incluir o contexto [local-extensions]
; para possibilitar que um ramal chame outro ou o D-Monitor:
;
include => local-extensions
;
; Agora, vamos incluir a seqüência do contexto [always-out-pots], que é
; a que possibilita que cada ramal interno do nosso PBX efetue chamadas
; através da linha analógica (POTS):
;
include => always-out-pots
;
; Como usual, colocamos as extensões 'h' e 'i' com seu devido tratamento:
;
exten => h,1,Hangup
exten => i,1,Congestion
exten => i,2,Hangup
;
;
;
[always-out-pots]
;
; Aqui, colocamos as ligações que sempre irão para a linha
; externa analógica (POTS).
;
; Sempre que o usuário de um ramal interno digitar um número que
; se encaixe numa das configurações abaixo, o Asterisk enviará
; a licação para o canal Zap/1 (configurado para linha analógica).
;
; Abaixo: números de qualquer tamanho com o zero no início correspondem
; a ligações para fora da USP (que sempre têm o zero no início, indicando
```

```
; solicitação à linha externa). Assim, basta discarmos no canal Zap/1
; o mesmo número digitado pelo usuário do ramal, que vem na
; variável EXTEN:
;
exten => _0X.,1,Dial(Zap/1/${EXTEN})
exten => _0X.,2,Congestion
exten => _0X.,3,Hangup
exten => _0X.,102,Congestion
exten => _0X.,103,Hangup
;
;
; Mesma coisa que acima, mas com '9' como indicador de chamada externa:
; Observe que, como se trata de chamada externa, discamos o zero antes
; de EXTEN, e tiramos o '9' inicial através do artifício ":1", resultando
; 0${EXTEN:1}.
;
exten => _9X.,1,Dial(Zap/1/0${EXTEN:1})
exten => _9X.,2,Congestion
exten => _9X.,3,Hangup
exten => _9X.,102,Congestion
exten => _9X.,103,Hangup
;
;
; Abaixo, coletamos as ligações para ramais da USP, que só têm
; 4 dígitos. O '1' no início será o indicador para ligações
; para ramais da USP.
;
exten => _1XXXX,1,Dial(Zap/1/${EXTEN:1})
exten => _1XXXX,2,Congestion
exten => _1XXXX,3,Hangup
exten => _1XXXX,102,Congestion
exten => _1XXXX,103,Hangup
;
;
;
[local-extensions]
;
; Este é o contexto que contém a seqüência de tratamento de ligações
```

```
; dirigidas a algum ramal interno do sistema (incluindo o D-Monitor).
;
exten => _ZXXX,1,Macro(internalsip,${EXTEN}) ; Z = qualquer dígito
exten => _ZXXX,2,Goto(0,1)                   ;     de 1 a 9
;
exten => 0,1,Hangup
;
;
;
[macro-internalsip]
;
; Define o comportamento padrão para as chamadas a ramais internos (SIP).
;
exten => s,1,GotoIf($[${ARG1} = 0]?s,100:)
exten => s,2,GotoIf($[${ARG1} = ${RAMAL_DMONITOR}]?d-monitor,s,1:)
exten => s,3,Dial(SIP/${ARG1},${TEMPO_RAMAL})
exten => s,4,Goto(s-${DIALSTATUS},1)
exten => s-CHANUNAVAIL,1,Playback(${SOUND_DIR}/ramal_nao_existente)
;exten => s-CHANUNAVAIL,2,Playback(${SOUND_DIR}/chamando_ramal_padrao)
;exten => s-CHANUNAVAIL,2,Playback(${SOUND_DIR}/ate_logo)
;exten => s-CHANUNAVAIL,3,Hangup
;exten => s-CHANUNAVAIL,3,Macro(internalsip,${RAMAL_PADRAO})
exten => s-NOANSWER,1,Playback(${SOUND_DIR}/ramal_indisponivel)
exten => s-NOANSWER,2,Playback(${SOUND_DIR}/ate_logo)
exten => s-NOANSWER,3,Hangup
exten => s-BUSY,1,Playback(${SOUND_DIR}/ramal_indisponivel)
exten => s-BUSY,2,Playback(${SOUND_DIR}/ate_logo)
exten => s-BUSY,3,Hangup
exten => _s.,1,Goto(s-NOANSWER,1)
;
exten => s,100,Hangup
;
;
[from-analog]
;
; Este é o contexto para onde as ligações provenientes da linha externa
; são enviadas (veja o arquivo 'zapata.conf').
;
```

```
; Configuramos o Asterisk para esperar a linha chamar por um período
; especificado em TEMPO_CHAMADA (ver seção 'globals', no início
; deste arquivo).
;
; Espera o telefone normal chamar por algum tempo:
;
exten => s,1,WaitForRing(${TEMPO_CHAMADA})
;
; Se ninguém no Laboratório atendeu, o Asterisk atende:
;
exten => s,2,Answer()
;
; Ajusta os limites de tempo de digitação:
;
exten => s,3,DigitTimeout(${TEMPO_MAX_DIGITO})
exten => s,4,ResponseTimeout(${TEMPO_MAX_RESPOSTA})
;
; Reproduz a mensagem padrão de boas-vindas:
;
exten => s,5,Macro(greetings)
;
; Solicita a digitação do ramal desejado:
;
exten => s,6,Background(${SOUND_DIR}/digite_o_ramal)
;
; Estabelecemos o zero como o encerrador imediato do sistema do Asterisk.
;
exten => 0,1,Hangup
;
; Envia qualquer número de 4 algarismos para a chamada aos ramais
; internos:
;
exten => _ZXXX,1,Macro(internalsip,${EXTEN})
exten => _ZXXX,2,Goto(s,6)
;
; No caso de timeout (tempo sem que o usuário digite o número),
; chama o ramal padrão:
;
```

```
exten => t,1,Playback(${SOUND_DIR}/chamando_ramal_padrao)
exten => t,2,Macro(internalsip,${RAMAL_PADRAO})
exten => t,3,Playback(${SOUND_DIR}/ate_logo)
exten => t,4,Hangup
;
; As extensões i (invalid) e h (hangup) devem constar em todos os
; contextos que manipulam chamadas diretamente, para o encerramento
; apropriado da ligação em caso de desligamento da outra parte ou da
; digitação de um ramal inválido.
;
exten => i,1,Wait(1)
exten => i,2,Playback(${SOUND_DIR}/ramal_nao_existente)
exten => i,3,Playback(${SOUND_DIR}/chamando_ramal_padrao)
exten => i,4,Goto(t,2)
;
exten => h,1,Hangup
;
;
;
[d-monitor]
;
; *** Contexto de entrada para o D-Monitor ***
;
; Solicita o número do usuário (senha) e, se correta, direciona o
; usuário para o menu inicial. No caso de três tentativas erradas
; na digitação da senha, encerra a ligação.
;
exten => s,1,SetVar(MYCOUNT=0)
exten => s,2,Background(${SOUND_DIR}/bem_vindo_dm)
exten => s,3,Background(${SOUND_DIR}/digite_numero_usuario)
;
exten => _XXXX,1,GotoIf($[${EXTEN} = ${NUM_USUARIO_DM}]?dm-menu,s,1:i,1)
;
exten => i,1,Playback(${SOUND_DIR}/nao_cadastrado)
exten => i,2,GotoIf($[${MYCOUNT} < 3]?:t,1)
exten => i,3,SetVar(MYCOUNT = $[${MYCOUNT} + 1])
exten => i,4,Goto(s,3)
;
```

```
exten => t,1,Playback(${SOUND_DIR}/ate_logo)
exten => t,2,Hangup
;
exten => h,1,Hangup
;
;
[dm-menu]
;
; *** Entrada para o menu do D-Monitor ***
;
exten => s,1,Macro(dm-inicializa)
exten => s,2,Wait(1)
exten => s,3,DBget(ATIVADO=dmonitor/ativado)
exten => s,4,GotoIf($[${ATIVADO} = 1]?:s,10)
exten => s,5,Playback(${SOUND_DIR}/monitor_ativado)
exten => s,6,Goto(dm-menu-ativado,s,1)
;
exten => s,10,Playback(${SOUND_DIR}/monitor_desativado)
exten => s,11,Goto(dm-menu-desativado,s,1)
;
; se ocorreu algum erro no DBget:
;
exten => s,104,Playback(${SOUND_DIR}/erro_durante_a_operacao)
exten => s,105,Playback(${SOUND_DIR}/ate_logo)
exten => s,106,Hangup
;
;
[dm-menu-ativado]
;
; *** Menu para o caso de monitoramento ativado ***
;
; Reproduz mensagens com as opções do menu:
;
exten => s,1,Macro(dm-numero-invasoes)
exten => s,2,Background(${SOUND_DIR}/para_desativar)
exten => s,3,Background(${SOUND_DIR}/info_ultima_invasao_3)
exten => s,4,Background(${SOUND_DIR}/limpar_dados_4)
exten => s,5,Background(${SOUND_DIR}/para_encerrar)
```

```
;
; colocar as opções:
;
exten => 0,1,Playback(${SOUND_DIR}/ate_logo)
exten => 0,2,Hangup
;
exten => 2,1,Macro(dm-desativa-monitoramento)
exten => 2,2,Goto(s,2)
;
exten => 3,1,Macro(dm-info-invasao)
exten => 3,2,Goto(s,2)
;
exten => 4,1,Macro(dm-limpa-dados)
exten => 4,2,Goto(s,2)
;
exten => t,1,Playback(${SOUND_DIR}/ate_logo)
exten => t,2,Hangup
;
exten => i,1,Playback(${SOUND_DIR}/opcao_invalida)
exten => i,2,Goto(s,2)
;
exten => h,1,Hangup
;
;
[dm-menu-desativado]
;
; *** Menu para o caso de monitoramento desativado ***
;
exten => s,1,Macro(dm-numero-invasoes)
exten => s,2,Background(${SOUND_DIR}/para_ativar)
exten => s,3,Background(${SOUND_DIR}/info_ultima_invasao_3)
exten => s,4,Background(${SOUND_DIR}/limpar_dados_4)
exten => s,5,Background(${SOUND_DIR}/para_encerrar)
;
; colocar as opções:
;
exten => 0,1,Playback(${SOUND_DIR}/ate_logo)
exten => 0,2,Hangup
```

```
;
exten => 1,1,Macro(dm-ativa-monitoramento)
exten => 1,2,Goto(s,2)
;
exten => 3,1,Macro(dm-info-invasao)
exten => 3,2,Goto(s,2)
;
exten => 4,1,Macro(dm-limpa-dados)
exten => 4,2,Goto(s,2)
;
exten => t,1,Playback(${SOUND_DIR}/ate_logo)
exten => t,2,Hangup
;
exten => i,1,Playback(${SOUND_DIR}/opcao_invalida)
exten => i,2,Goto(s,2)
;
exten => h,1,Hangup
;
;
;
[dm-invasao-detectada]
;
;
exten => s,1,SetVar(COUNT=0)
exten => s,2,DBget(ATIVADO=dmonitor/ativado)
exten => s,3,GotoIf($[${ATIVADO} = 0]?h,1:)
exten => s,4,DigitTimeout(45)
exten => s,5,ResponseTimeout(45)
exten => s,6,Macro(dm-registra-invasao)
exten => s,7,Background(${SOUND_DIR}/bem_vindo_dm)
exten => s,8,Background(${SOUND_DIR}/digite_numero_usuario)
exten => s,9,WaitExten(10)
exten => s,10,SetVar(COUNT=$[${COUNT} + 1])
exten => s,11,GotoIf($[${COUNT} > 3]?h,1:s,7)
;
exten => s,103,DBput(dmonitor/ativado=0)
exten => s,104,Goto(h,1)
;
```

```
exten => _ZXXX,1,DigitTimeout(${TEMPO_MAX_DIGITO})
exten => _ZXXX,2,ResponseTimeout(${TEMPO_MAX_RESPOSTA})
exten => _ZXXX,3,GotoIf($[${EXTEN} = ${NUM_USUARIO_DM}]?dm-menu,s,1:)
exten => _ZXXX,4,Playback(${SOUND_DIR}/nao_cadastrado)
exten => _ZXXX,5,Goto(s,8)
;
exten => failed,1,Macro(dm-registra-invasao)
exten => failed,2.Hangup
;
exten => 0,1,Hangup
;
exten => i,1,Playback(${SOUND_DIR}/nao_cadastrado)
exten => i,2,Goto(s,8)
;
exten => h,1,Hangup
;
;
;
[macro-greetings]
;
; Macro para a reprodução da mensagem de boas-vindas.
;
exten => s,1,Wait(1)
exten => s,2,Background(${SOUND_DIR}/design_lab)
exten => s,3,GotoIfTime(00:00-11:59,*,*,*?s,10)
exten => s,4,GotoIfTime(12:00-17:59,*,*,*?s,12)
exten => s,5,GotoIfTime(18:00-23:59,*,*,*?s,14)
exten => s,10,Background(${SOUND_DIR}/bom_dia)
exten => s,11,Goto(s,20)
exten => s,12,Background(${SOUND_DIR}/boa_tarde)
exten => s,13,Goto(s,20)
exten => s,14,Background(${SOUND_DIR}/boa_noite)
exten => s,15,Goto(s,20)
exten => s,20,Wait(0.5)
;
;
[macro-goodbye]
;
```

```
; Macro para mensagem de despedida
;
exten => s,1,Playback(${SOUND_DIR}/ate_logo)
exten => s,2,Hangup
;
;
;
[macro-dm-inicializa]
;
; *** Macro para inicialização das variáveis do D-Monitor ***
;
exten => s,1,DBget(TEMP=dmonitor/ativado)
exten => s,2,Goto(10)
;
exten => s,10,DBget(INVASOES=dmonitor/invasoes)
;
; Se não encontrou a chave dmonitor/ativado,
; cria-a com valor 0:
;
exten => s,102,DBput(dmonitor/ativado=0)
exten => s,103,Goto(10)
;
; Se não encontrou a chave dmonitor/invasoes,
; cria-a com valor 0:
;
exten => s,111,DBput(dmonitor/invasoes=0)
;
;
;
[macro-dm-registra-invasao]
;
exten => s,1,DBget(INVASOES=dmonitor/invasoes)
exten => s,2,DBput(dmonitor/invasoes=$[${INVASOES} + 1])
exten => s,3,Macro(pega-hora-atual)
exten => s,4,DBput(dmonitor/invasao_hora=${HOUR})
exten => s,5,DBput(dmonitor/invasao_minuto=${MINUTE})
exten => s,6,Macro(pega-dia-semana)
exten => s,7,DBput(dmonitor/invasao_dia=${WEEKDAY})
```

```
;
;
exten => s,102,SetVar(INVASOES=0)
exten => s,103,Goto(2)
;
;
;
[macro-dm-numero-invasoes]
;
; Informa o número de invasões registradas
;
exten => s,1,DBget(NUM=dmonitor/invasoes)
exten => s,2,GotoIf($[${NUM} = 0]?s,102:)
exten => s,3,Playback(${SOUND_DIR}/foram_registradas)
exten => s,4,Macro(dm-diz-numero-feminino,${NUM})
exten => s,5,GotoIf($[${NUM} = 1]?10:)
exten => s,6,Playback(${SOUND_DIR}/novas_invasoes)
;
exten => s,10,Playback(${SOUND_DIR}/nova_invasao)
;
exten => s,102,Playback(${SOUND_DIR}/nao_ha_invasoes)
exten => s,103,DBput(dmonitor/invasoes=0)
;
;
;
[macro-dm-diz-numero-feminino]
;
; Macro para pronunciar um número de 1 a 9, no feminino.
;
; Observe que, a partir da prioridade 21, as prioridades
; são fornecidas em números não-consecutivos.
; Isto faz com que, uma vez executada alguma dessas prioridades,
; a macro encerre e retorne à prioridade que a chamou.
;
exten => s,1,GotoIf($[${ARG1} = 1]?s,21:)
exten => s,2,GotoIf($[${ARG1} = 2]?s,23:)
exten => s,3,GotoIf($[${ARG1} = 3]?s,25:)
exten => s,4,GotoIf($[${ARG1} = 4]?s,27:)
```

```
exten => s,5,GotoIf($[${ARG1} = 5]?s,29:)
exten => s,6,GotoIf($[${ARG1} = 6]?s,31:)
exten => s,7,GotoIf($[${ARG1} = 7]?s,33:)
exten => s,8,GotoIf($[${ARG1} = 8]?s,35:)
exten => s,9,GotoIf($[${ARG1} = 9]?s,37:)
exten => s,10,GotoIf($[${ARG1} > 9]?s,39:)
;
exten => s,21,Playback(${SOUND_DIR}/uma)
;
exten => s,23,Playback(${SOUND_DIR}/duas)
;
exten => s,25,Playback(${SOUND_DIR}/tres)
;
exten => s,27,Playback(${SOUND_DIR}/quatro)
;
exten => s,29,Playback(${SOUND_DIR}/cinco)
;
exten => s,31,Playback(${SOUND_DIR}/seis)
;
exten => s,33,Playback(${SOUND_DIR}/sete)
;
exten => s,35,Playback(${SOUND_DIR}/oito)
;
exten => s,37,Playback(${SOUND_DIR}/nove)
;
exten => s,39,Playback(${SOUND_DIR}/mais_de)
exten => s,40,Playback(${SOUND_DIR}/nove)
;
;
;
[macro-dm-info-invasao]
;
exten => s,1,DBget(INVASOES=dmonitor/invasoes)
exten => s,2,GotoIf($[${INVASOES} < 1]?102:)
exten => s,3,DBget(INVASAO_HORA=dmonitor/invasao_hora)
exten => s,4,DBget(INVASAO_MINUTO=dmonitor/invasao_minuto)
exten => s,5,DBget(INVASAO_DIA=dmonitor/invasao_dia)
exten => s,6,Playback(${SOUND_DIR}/ultima_invasao)
```

```
exten => s,7,Macro(diz-dia-semana,${INVASAO_DIA})
exten => s,8,Playback(${SOUND_DIR}/as)
exten => s,9,SayNumber(${INVASAO_HORA})
exten => s,10,Playback(${SOUND_DIR}/horas)
exten => s,11,Playback(${SOUND_DIR}/e)
exten => s,12,SayNumber(${INVASAO_MINUTO})
exten => s,13,Playback(${SOUND_DIR}/minutos)
;
exten => s,102,Playback(${SOUND_DIR}/nao_ha_invasoes)
;
exten => s,104,Goto(102)
exten => s,105,Goto(102)
exten => s,106,Goto(102)
;
;
;
[macro-pega-hora-atual]
;
; Eis uma macro para pegar a hora atual e armazená-la em variáveis.
; Armazena a hora em 'HOUR' e os minutos em 'MINUTE'.
;
; DDMMYYYY-HH:MM:SS
;
exten => s,1,SetVar(HOUR=${DATETIME:9:2})
exten => s,2,SetVar(MINUTE=${DATETIME:12:2})
;
;
;
[macro-pega-hora-atual2]
;
exten => s,1,SetVar(TRY=1)
exten => s,2,SetVar(CHOUR=23)
exten => s,3,GotoIfTime(${CHOUR}:00-${CHOUR}:59,*,*,*?s,10)
exten => s,4,GotoIf($[${CHOUR} = 0]?s,50:)
exten => s,5,SetVar(CHOUR=$[${CHOUR} - 1])
exten => s,6,Goto(s,3)
;
exten => s,10,SetVar(CMINUTE=59)
```

```
exten => s,11,GotoIfTime(${CHOUR}:${CMINUTE},*,*,*?s,20)
exten => s,12,GotoIf($[${CMINUTE} = 0]?s,50:)
exten => s,13,SetVar(CMINUTE=$[${CMINUTE} - 1])
exten => s,14,Goto(s,11)
;
exten => s,20,SetGlobalVar(HOUR=${CHOUR})
exten => s,21,SetGlobalVar(MINUTE=${CMINUTE})
;
exten => s,50,GotoIf($[${TRY} > 3]?s,60:)
exten => s,51,SetVar(TRY=$[${TRY} + 1])
exten => s,52,Goto(s,2)
;
; Não conseguiu obter a hora atual:
;
exten => s,60,SetGlobalVar(HOUR=-1)
exten => s,61,SetGlobalVar(MINUTE=-1)
;
;
[macro-pega-dia-semana]
;
; Acha o dia da semana atual e armazena na variável global 'WEEKDAY'.
;
exten => s,1,GotoIfTime(*,sun,*,*?s,21)
exten => s,2,GotoIfTime(*,mon,*,*?s,23)
exten => s,3,GotoIfTime(*,tue,*,*?s,25)
exten => s,4,GotoIfTime(*,wed,*,*?s,27)
exten => s,5,GotoIfTime(*,thu,*,*?s,29)
exten => s,6,GotoIfTime(*,fri,*,*?s,31)
exten => s,7,GotoIfTime(*,sat,*,*?s,33)
exten => s,8,SetGlobalVar(WEEKDAY=-1)
;
exten => s,21,SetVar(CDAY=1)
exten => s,22,Goto(s,50)
exten => s,23,SetVar(CDAY=2)
exten => s,24,Goto(s,50)
exten => s,25,SetVar(CDAY=3)
exten => s,26,Goto(s,50)
exten => s,27,SetVar(CDAY=4)
```

```
exten => s,28,Goto(s,50)
exten => s,29,SetVar(CDAY=5)
exten => s,30,Goto(s,50)
exten => s,31,SetVar(CDAY=6)
exten => s,32,Goto(s,50)
exten => s,33,SetVar(CDAY=7)
exten => s,34,Goto(s,50)
;
exten => s,50,SetGlobalVar(WEEKDAY=${CDAY})
;
;
;
[macro-diz-dia-semana]
;
; Macro para pronunciar o dia da semana passado como parâmetro.
; (1=domingo, 2=segunda-feira, ..., 7=sábado)
;
exten => s,1,GotoIf($[${ARG1} = 1]?s,11:)
exten => s,2,GotoIf($[${ARG1} = 2]?s,13:)
exten => s,3,GotoIf($[${ARG1} = 3]?s,15:)
exten => s,4,GotoIf($[${ARG1} = 4]?s,17:)
exten => s,5,GotoIf($[${ARG1} = 5]?s,19:)
exten => s,6,GotoIf($[${ARG1} = 6]?s,21:)
exten => s,7,GotoIf($[${ARG1} = 7]?s,23:)
exten => s,8,Goto(dm-erro,s,1)
;
exten => s,11,Playback(${SOUND_DIR}/domingo)
;
exten => s,13,Playback(${SOUND_DIR}/segunda_feira)
;
exten => s,15,Playback(${SOUND_DIR}/terca_feira)
;
exten => s,17,Playback(${SOUND_DIR}/quarta_feira)
;
exten => s,19,Playback(${SOUND_DIR}/quinta_feira)
;
exten => s,21,Playback(${SOUND_DIR}/sexta_feira)
;
```

```
exten => s,23,Playback(${SOUND_DIR}/sabado)
;
;
;
[macro-pega-e-diz-dia-semana]
;
exten => s,1,Macro(pega-dia-semana)
exten => s,2,Macro(diz-dia-semana,${WEEKDAY})
;
;
;
[dm-erro]
;
exten => s,1,Playback(${SOUND_DIR}/erro_durante_a_operacao)
exten => s,2,Playback(${SOUND_DIR}/ate_logo)
exten => s,3,Hangup
```

/home/kurumin/dmonitor/Invasao_Detectada.call

```
#
# Invasao_Detectada.call - Arquivo que aciona uma chamada automática
# do Asterisk. Direciona para o contexto [dm-invasao-detectada] em
# 'extensions.conf'.
# O Asterisk efetua automaticamente a ligação assim que este
# arquivo é colocado no diretório /var/spool/asterisk/outgoing
# Autor: José Adriano Meireles Pardi - março de 2006
#
Channel: Zap/1/091513977
MaxRetries: 0
Context: dm-invasao-detectada
Extension: s
Priority: 1
```

/home/kurumin/dmonitor/scripts/verifica_intrusao.sh

```
#!/bin/bash
# Este script é executado a cada minuto através do cron do
# Linux. Verifica se chegou o arquivo de imagem pelo FTP
# e, em caso positivo, aciona o Asterisk por meio de
# mover o arquivo InvasaoDetectada.call para o
# diretório /var/spool/asterisk/outgoing
# Autor: José Adriano Meireles Patrdi - março de 2006
#
cd /home/kurumin/ftp
file='01.jpg'
if [ -e $file ]
then
   rm $file
   cd /home/kurumin/dmonitor
   cp Invasao_Detectada.call /var/spool/asterisk
   cd /var/spool/asterisk
   mv Invasao_Detectada.call outgoing
fi
```

Arquivos de som com as mensagens usadas pelo sistema:

Todos estão em formato WAV, e foram gravados com padrão PCM de 8 kHz, 16 bits por amostragem. Estão em: /usr/share/asterisk/sounds/DMonitor

as.wav
ate_logo.wav
aviso_dm_invasao_detect.wav
bem_vindo_dm.wav
boa_noite.wav
boa_tarde.wav
bom_dia.wav
chamando_ramal_padrao.wav
cinco.wav
design_lab.wav
digite_numero_usuario.wav
digite_o_ramal.wav
dois.wav
domingo.wav
duas.wav
e_meia.wav
erro_durante_a_operacao.wav
e.wav
foram_detectadas.wav
grave_sua_mensagem.wav
ha_aproximadamente.wav
ha_mais_de.wav
horario.wav
horas.wav
hora.wav
info_novas_invasoes_3.wav
info_ultima_invasao_3.wav
invasao_detectada.wav
limpar_dados_4.wav
mais_de.wav
menos_de.wav
minutos.wav
monitor_ativado.wav
monitor_desativado.wav
nao_cadastrado.wav
nao_ha_invasoes.wav
no_momento_digite_1.wav
no_momento_nao_podemos.wav
no_momento_obrigado.wav
nova_invasao.wav
novas_invasoes.wav
nove.wav
numero_desejado.wav
oito.wav
para_ativar.wav
para_desativar.wav
para_encerrar_v.wav
para_encerrar.wav
quarta_feira.wav
quatro.wav
quinta_feira.wav
ramal_indisponivel.wav
ramal_nao_existente.wav
ramal_nao_exist.wav
sabado.wav
segunda_feira.wav
seis.wav
sete.wav
sexta_feira.wav
tchau.wav
terca_feira.wav
tres.wav
ultima_invasao.wav
uma.wav
um.wav
usuario_nao_cadastrado.wav