

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA

FAKULTA PROVOZNĚ EKONOMICKÁ

Obor informatika



BAKALÁŘSKÁ PRÁCE

Téma: Použití VoIP pomocí open source Asterisk PBX

Vypracoval: Lukáš Kaucký

Vedoucí bakalářské práce: Ing. David Buchtela Ph.D.

Čestné prohlášení

Prohlašuji, že jsem bakalářskou práci na téma **Použití VoIP pomocí open source Asterisk PBX** zpracoval samostatně pod vedením vedoucího bakalářské práce pana Ing. Davida Buchtely Ph.D za použití literatury uvedené v seznamu na konci práce. Jako autor této bakalářské práce prohlašuji, že jsem v souvislosti s jejím vytvořením vědomě neporušil autorská práva třetích osob. Zároveň dávám výslovný souhlas k použití této práce provozně ekonomické fakultě české zemědělské univerzity pro účely výuky nebo vlastního použití.

V Praze dne 15. 3. 2013

.....

Lukáš Kaucký

PODĚKOVÁNÍ

Děkuji tímto panu Ing. Davidovi Buchtelovi Ph.D. za odborné vedení a rady při zpracování bakalářské práce. Velké díky patří mému zaměstnavateli Štěpánovi Pokornému, majiteli firmy Altus software s.r.o., za poskytnutí prostředků potřebných pro zpracování mé práce a za jeho podporu a toleranci mé časté nepřítomnosti v zaměstnání. Největší díky však patří mému otci Karlovi Kauckému, jenž mne ve studiu od počátku podporoval, ale této práce se již bohužel nedožil. Nakonec bych rád poděkoval i osobám, které se o mé vděčnosti nikdy nedozví a to panu Mark Spencer, autorovi telefonní ústředny Asterisk, pánům Leif Madsen, Jim Van Meggelen a Russel Bryant, autorům knihy [1] a ostatním lidem, kteří na internetových diskusích dokázali pomoci ostatním s problémy, které se při nasazení telefonní ústředny Asterisk vyskytly.

Souhrn

Bakalářská práce se zabývá problematikou použití technologie VoIP za pomoci open source ústředny Asterisk. Ve své teoretické části popisuje za pomoci uvedených zdrojů jednotlivé komunikační protokoly související s implementací VoIP ve firemním prostředí, zvláště se pak zaměřuje na princip a problémy protokolu SIP. Zároveň nabízí návod na způsob určení hardwarových a softwarových požadavků pro provoz telefonní ústředny.

V praktické části se pak zabývá vzorovou instalací a obecně použitelným nastavením telefonní ústředny. Jsou zde popsány základní parametry a funkce ústředny použité ve vzorové konfiguraci. Největší pozornost je věnována nastavení parametrů klientů, kteří se mohou k ústředně připojit prostřednictvím protokolu SIP, a vytáčecímu plánu, který řídí způsob zpracování hovorů. V závěrečné části jsou uvedeny parametry nutné pro připojení telefonů k ústředně a základní příkazy pro ovládání chodu telefonní ústředny.

Výsledkem práce je obecně použitelný návod na konfiguraci telefonní ústředny Asterisk. Tato práce může pomoci IT administrátorům zavést v jejich podniku IP telefonii, aniž by měli předchozí zkušenosti s implementací VoIP. Vzorová instalace ústředny ve formě virtuálního disku, připravená po drobných úpravách parametrů k okamžitému použití, je součástí elektronické přílohy na DVD.

Klíčová slova: asterisk, ústředna, SIP, VoIP, vytáčecí plán, telefon, kodek, audio formát, kontext, DTMF

Summary

The bachelor thesis deals with the use of VoIP using Asterisk open source PBX. The theoretical part describes using the sources listed each communications protocol associated with the implementation of VoIP in a business environment, particularly focusing on the principles and problems of the SIP protocol. It also offers guidance on how to determine hardware and software requirements for running the telephone exchange.

The practical part is being focused on model generally applicable installation and setup PBX. There are described basic PBX functions and parameters used in the model configuration. Most attention is paid to setting the parameters of clients who can connect to the PBX via SIP, and dial plan that controls how to handle calls. The final section lists the parameters required to connect phones to the PBX and basic commands for controlling the operation of the telephone exchange.

The result of this work is generally applicable guidance on configuring Asterisk PBX. This work can help IT administrators to implement in their company IP telephony, without having had prior experience with the implementation of VoIP. Sample PBX installation as a virtual disk, ready after minor adjustments for immediate use, is part of the electronic attachments on DVD.

Keywords: asterisk, PBX, SIP, VoIP, dial plan, phone, codec, audio format, context, DTMF

Obsah

Obsah	7
1. Úvod.....	9
2. O Asterisk PBX	9
3. Popis základních komunikačních protokolů	10
3.1 TCP	10
3.2 UDP	11
3.3 NAT	11
3.4 RTP	12
3.5 SIP.....	12
3.6 IAX2	17
4. Audio formát G.711	18
5. Požadavky na hardware, operační systém a konektivitu	19
5.1 Požadavky na hardware	19
5.2 Požadavky na operační systém	21
5.3 Požadavky na konektivitu	22
6. Vlastní návrh řešení	24
6.1 Instalace	24
6.1.1 Příprava virtualizačního prostředí pro vzorovou instalaci.....	24
6.1.2 Instalace telefonní ústředny Asterisk.....	27
6.2 Nastavení telefonní ústředny.....	28
6.2.1 Nastavení modulů zaváděných při startu	29
6.2.2 Nastavení protokolu SIP a koncových zařízení – sip.conf	30
6.2.3 Nastavení hlasové schránky – voicemail.conf.....	40
6.2.4 Nastavení fronty volajících – queues.conf.....	40
6.2.5 Vytvoření audio souborů s oznámeními	42
6.2.6 Nastavení vytáčecího plánu – extensions.conf	43
6.2.7 Syntaxe základních operátorů a funkcí.....	44
6.2.8 Popis logiky vzorového vytáčecího plánu	49
6.3 Nastavení telefonů, základní příkazy CLI ústředny.....	57

6.3.1 Nastavení telefonů	57
6.3.2 Základní příkazy CLI ústředny	57
7. Závěr	59
8. Přílohy.....	61
8.1 Příloha č. 1 - základní instalace FreeBSD 9.1	61
8.2 Příloha č. 2 - konfigurace systému a instalace dalších součástí.....	69
8.3 Příloha č. 3 – konfigurační soubor modules.conf	70
8.4 Příloha č. 4 – konfigurační soubor /etc/rc.conf.....	71
8.5 Příloha č. 5 – kompletní soubor sip.conf	72
8.6 Příloha č. 6 - kompletní soubor extensions.conf.....	72
9. Obsah elektronické přílohy	76
Použité zdroje:	77
Seznam tabulek a obrázků	79

1. Úvod

Za počátek technologie VoIP neboli Voice Over IP (Přenos hlasu pomocí protokolu IP) je podle Jiřího Peterky všeobecně považován rok 1995 [3]. Od té doby tato technologie prošla značným vývojem a dospěla do stavu, kdy je schopna plně nahradit klasickou telefonii.

Stabilní telefonní spojení je ve firemním prostředí nezbytnou součástí komunikační infrastruktury. Cílem této práce je nabídnout řešení firemní telefonie pomocí open source telefonní ústředny Asterisk PBX (<http://www.asterisk.org/>), jejíž vývoj je sponzorován firmou Digium (<http://www.digium.com/>). Tato práce nabízí uživateli kromě teoretických základů také popis instalace a základního nastavení telefonní ústředny.

2. O Asterisk PBX

Historie projektu Asterisk PBX (dále v tomto dokumentu již pouze Asterisk nebo ústředna) sahá až do roku 1999 ([1] - *strana 22 - Foreword XX*), kdy Mark Spencer¹ uvolnil základní kód ústředny pod GPL open source licencí. Po mnoha letech vývoje (k lednu 2013 je aktuální verze 11.1.2) se za účasti tisíců vývojářů stal z telefonní ústředny, která uměla pouze spojovat telefonní hovory, nástroj poskytující nejen běžné přepojování hovorů, ale i jiné užitečné funkce, jako jsou nahrávání hovorů, záznamník, fax, zaznamenávání historie hovorů do databáze, fronty volajících (např. pro technickou podporu), zpětné volání, použití služby ENUM, videokonference a další.

V našem případě se budeme zabývat verzí 1.8, jež je předmětem publikace Asterisk: the definitive guide [1] a jež je i v kolekci stabilních portů² operačního systému FreeBSD 9.0 release³, na kterém bude vytvořena vzorová konfigurace.

¹ Zakladatel firmy Digium®, Inc.

² kolekce instrukcí pro pohodlné nainstalování software z originálních zdrojových kódů [5]

³ <http://www.freebsd.cz/cs/>

3. Popis základních komunikačních protokolů

Princip VoIP je v podstatě jednoduchý. Hlas jednoho uživatele zaznamenaný mikrofonem telefonu je převeden do digitální podoby, pomocí IP (Internet Protocol) přenesen do cíle a tam opět převeden na zvuk, který slyší druhý uživatel ve sluchátku telefonu. Pro přenos těchto dat se používají speciální signalizační protokoly, jejichž úkolem je vzájemná domluva dvou koncových zařízení mezi sebou o tom, jak se budou hlasová data přenášet a protokoly, jež mají za úkol přenášet data samotná.

V tomto dokumentu se setkáme s protokoly SIP, IAX2 a RTP, které Asterisk (kromě jiných) nejčastěji používá pro komunikaci s koncovými zařízeními a ostatními ústřednami. Vzhledem k tomu, že v popisu protokolů budou zmíněny i termíny TCP, UDP a NAT, je vhodné je krátce popsat právě zde.

3.1 TCP

TCP (Transmission Control Protocol) [2], [9] představující transportní vrstvu, je spolehlivá spojovaná služba postavená nad protokolem IP. Dvě konkrétní aplikace na jednom či dvou různých koncových zařízeních spolu naváží spojení a vytvoří na dobu spojení takzvaný virtuální okruh. Protokol zajišťuje spolehlivé doručování dat ve správném pořadí. Jednotlivé odesílané bloky dat (pakety) jsou číslovány pořadovým číslem, a dokud není doručena jedna (předchozí) část, nemohou být doručeny části následující. Pokud dojde během přenosu ke ztrátě nebo poškození paketu, není tento paket příjemcem potvrzen a odesílatel jej posílá znovu do té doby, dokud není jeho úspěšný příjem potvrzen příjemcem. To znamená, že TCP není vhodný pro přenos proudových dat, jako je např. zvuk, u něhož je preferováno včasné doručení před kompletností a správným pořadím. Pro přenos dat aplikací běžících v reálném čase je vhodnější protokol UDP.

TCP používá k rozlišení vzájemně komunikujících aplikací čísla portů, jež jsou obvykle přirovnávána k jednotlivým poštovním schránkám v domě na dané adrese. IP adresou je tedy rozuměn konkrétní dům a portem konkrétní poštovní schránka. Číslo portu je 16bitové kladné číslo. Používá se rozsah 1–65535. Číslo portu 0 je rezervováno a

nepoužívá se, čísla 1–1023 jsou tzv. dobře známé porty (well known ports), pro jejichž použití jsou na systémech typu UNIX nutná oprávnění superuživatele (root) [10].

3.2 UDP

UDP (User Datagram Protocol) [2], [10] je jednodušší alternativou TCP. Jedná se o službu, u níž není zaručeno doručení bloků dat (datagramů) ani jejich pořadí (protokol je někdy poněkud zavádějícím způsobem označován jako nespolehlivý). Odesílatel odešle datagram příjemci a již se nestará o to, zda byl úspěšně doručen, což znamená, že data jsou odesílána bez prodlev, které mohou při použití TCP díky nutnosti potvrzování a případného znovudesílání paketů nastat. Z toho důvodu je tento protokol vhodný mimo jiné pro aplikace běžící v reálném čase používající proudový přenos dat (např. právě zvuk).

UDP používá stejně jako TCP čísla portů k rozlišení vzájemně komunikujících aplikací.

3.3 NAT

NAT (Network Address Translation), neboli překlad síťových adres, je proces, který mimo jiné (viz [6]) umožňuje sdílení jedné veřejné IP adresy na vnějším rozhraní směrovače více zařízeními ve vnitřní síti.

Vycházíme-li například z [7], pak to v praxi znamená, že zařízení ve vnitřní síti odešle do vnější sítě paket se svou zdrojovou (neveřejnou) IP adresou a portem. Směrovač podporující NAT potom přepíše zdrojovou IP adresu tohoto paketu na svou veřejnou IP adresu a zdrojový TCP nebo UDP port na port, jenž danému odesílajícímu zařízení přidělí. Směrovač si do své interní tabulky uloží informaci o tomto přidělení a v okamžiku, kdy přijde odpověď od vzdáleného počítače, přepíše směrovač na základě údajů ve své tabulce hlavičku paketu tak, aby obsahoval vnitřní adresu a původní port volajícího zařízení ve vnitřní síti. Směrovač pak přepošle tento paket ke zpracování volajícímu zařízení, které na daném portu tento paket očekává.

V případě, že směrovač přijme na vnějším rozhraní paket, pro který nemá odpovídající záznam v tabulce a není ani odpovědí na jeho vlastní požadavek, zjednodušeně řečeno jej zahodí, z čehož vyplývá, že se zařízeními ve vnitřní síti za NATem nelze navázat přímé

spojení. Pro protokol SIP je nemožnost přímého spojení se zařízeními za NATem problémem, jak bude popsáno níže v kapitole 3.5.

3.4 RTP

RTP (Real-time Transport Protocol) [8] slouží k přenosu proudových multimediálních dat, jako jsou audio a video. V případě IP telefonie slouží tedy především k přenosu hlasu. RTP se používá v kombinaci s RTCP (RTP Control Protocol), jenž má za úkol řídit RTP relaci a sledovat kvalitu datového toku, a v kombinaci se signalizačními protokoly, jako jsou SIP nebo H.323. RTP je postaven nad UDP (User Datagram Protocol⁴) a pro komunikaci používá tzv. vysoké porty 1024 až 65535 (Asterisk používá ve výchozím nastavení porty 10 000 až 20 000. Toto nastavení lze změnit v konfiguračním souboru *rtp.conf*), přičemž sudá čísla portu jsou určena pro RTP a nejbližší vyšší lichá pak pro RTCP. UDP je použit z důvodu požadavku na rychlé doručení dat bez ohledu na možné ztráty (viz kapitolu 3.2).

3.5 SIP

SIP (Session Initiation Protocol – protokol pro inicializaci relací) [1],[11],[12] je relativně jednoduchý protokol sloužící pro přenos signalizace v komunikačních relacích, jako jsou přenosy audia a videa pomocí IP. Veškerá komunikace mezi zařízeními probíhá ve výchozím nastavení přes UDP port 5060 pomocí jednoduchých textových příkazů, stejně jako například HTTP nebo SMTP. Základem SIP je, že každé koncové zařízení je rovnocenným členem relace, což znamená, že SIP nerozlišuje, zda je zařízení serverem či klientem. SIP nezprostředkovává přenos hlasu samotného, o to se stará již zmíněný protokol RTP (viz kapitolu 3.4).

SIP pouze zajišťuje následující činnosti [11],[12]:

- *Lokalizace účastníka* – nalezení spojení s koncovým zařízením
- *Zjištění stavu účastníka* – zjištění, zda je zařízení schopno navázat relaci (obsazeno, přesměrováno atd.)

⁴ http://en.wikipedia.org/wiki/User_Datagram_Protocol

- *Zjištění možností účastníka* – zjištění možností zařízení - typ kodeku (viz kapitolu 4), maximální přenosovou rychlost, typ přenosu (zda se jedná o audio či video) atd.
- *Vlastní navázání spojení* – nastavení parametrů RTP datového toku (čísla UDP portů, protokoly, kodeky) prostřednictvím SDP (Session Description Protocol)⁵ přenášeného v těle SIP paketu.
- *Řízení probíhajícího spojení* – případné změny vlastností v průběhu relace (změna IP adresy či portu, pozvání dalších účastníků a další) a činnosti spojené s jejím ukončením.

V případě běžné IP telefonie nenavazují obvykle zařízení komunikaci přímo mezi sebou, ale pomocí jednoho či více SIP proxy serverů, které obvykle plní i funkci SIP registrátora – v našem případě poskytovatele VoIP služeb.

SIP komunikace mezi dvěma účastníky registrovanými u různých operátorů může probíhat zjednodušeně například takto (viz také Tabulka 1 – *průběh hovoru mezi dvěma účastníky* na následující straně):

1.) Telefonista Weigel registrovaný u T-Systems vytočí číslo telefonisty Smoljaka.

T-Systems zjistí, že telefonista Smoljak u něj není registrován, ale ve své směrovací tabulce najde, že Smoljakovo číslo je registrováno u Coprosysu. T-Systems tedy kontaktuje Coprosys a předá mu požadavek. Coprosys nalezne Smoljaka ve své tabulce registrovaných účastníků a pokusí se spojit s jeho telefonem. Pokud je telefon zapnutý a není obsazený, začne vyzvánět a Coprosysu tuto skutečnost sdělí. Ten tuto zprávu předá zpět T-Systems a ten telefonu Weigela.

2.) Po nějaké době zvedne Smoljak svůj telefon. Ten pošle Coprosysu zprávu o svém zvednutí a ta se prostřednictvím operátorů dostane až k telefonu Weigela. Ten přijetí této informace potvrdí a potvrzení se opět dostane až k telefonu Smoljaka. V tom okamžiku může začít hovor.

⁵ http://cs.wikipedia.org/wiki/Session_Description_Protocol

3.) Oba telefonisté spolu hovoří a jejich digitalizovaný hlas je přenášen pomocí RTP, jehož parametry byly dohodnuty během navazování spojení.

4.) Po skončení hovoru telefonista Smoljak zavěsí a jeho telefon pošle prostřednictvím operátorů zprávu o ukončení hovoru telefonu Weigela. Ten přijetí této zprávy Smoljakovu telefonu potvrdí a hovor je ukončen.

Tabulka 1 – průběh hovoru mezi dvěma účastníky

<u>Telefon Weigela</u>	<u>T-Systems</u>	<u>Coprosys</u>	<u>Telefon Smoljaka</u>
1. Vytočení čísla telefonistou Weigelem			
INVITE	→	•	
•	←	<i>Trying</i>	
		INVITE	→→→→→
		•	←←←←←
			<i>Trying</i>
		INVITE	→
			<i>Vyzvánění</i>
<i>Vyzváněcí tón</i>	←	<i>Ringing</i>	←←←←←
		<i>Ringing</i>	←
			•
<hr/>			
2.		<i>Zvednutí telefonu telefonistou Smoljakem</i>	
<i>OK</i>	←	<i>OK</i>	←←←←←
		<i>OK</i>	←
			•
ACK	→	ACK	→→→→→
		ACK	→
			•
<hr/>			
3. Probíhá multimediální relace pomocí RTP (Telefonisté spolu hovoří).			
<hr/>			
4.		<i>Telefonista Smoljak zavěsil</i>	
•	←←←←←	←←←←←	BYE
OK	→→→→→	→→→→→	•

INVITE, **ACK** a **BYE** uvedené v tabulce jsou jedny ze základních metod SIP. **Trying**, **Ringing** a **OK** jsou jedny ze stavových hlášení. T-Systems a Coprosys jsou SIP registrátoři (VoIP operátoři).

Mezi základní metody SIP patří [11]:

REGISTER – registrace účastníka na serveru poskytovatele

INVITE – zahájení komunikace o plánované relaci

ACK – potvrzení zahájení relace

CANCEL – přerušení zahajování relace ještě před jejím navázáním

BYE – ukončení relace

OPTIONS – žádost o informace o možnostech volajícího bez sestavení volání

Chybová (stavová) hlášení jsou inspirována protokolem HTTP a jsou podobným způsobem rozdělena do těchto kategorií [11]:

1xx - průběh – krok probíhá bez problémů, ale ještě není ukončen

2xx - úspěch – krok byl ukončen bez problémů

3xx - přesměrování – krok probíhá, ale ještě se v souvislosti s ním něco očekává

4xx - chyba klienta – požadavek je chybný a nemůže být serverem zpracován

5xx - chyba serveru – požadavek je zřejmě v pořádku, ale chyba je na straně serveru

6xx - fatální chyba – zcela fatální chyba, kterou nelze jakkoliv zpracovat

Problémem pro komunikaci pomocí protokolů SIP a RTP je již zmíněný překlad adres (viz kapitolu 3.3), kdy zařízení (ústředna nebo telefon) jsou ve vnitřní síti a pomocí NAT sdílejí jednu veřejnou IP adresu s ostatními zařízeními. Nejčastějším případem je situace, kdy sice dojde k navázání spojení (telefony se kontaktují, volající slyší vyzváněcí tón a volanému zvoní telefon), ale po zvednutí sluchátka volaným není slyšet žádný zvuk. Příčina je v tom, že zařízení (směrovač, firewall) realizující překlad adres není schopen spárovat předchozí SIP komunikaci na portu 5060 s nově otevřenou RTP komunikací na portech, jež si zařízení mezi sebou dohodla, a tyto pakety zahazuje (příchozí RTP komunikace neodpovídá žádnému záznamu v tabulce přeložených adres).

Nejjednodušším řešením v případě použití ústředny Asterisk (což je náš případ) je otevření UDP portů 10 000 – 20 000 (výchozí nastavení) pro IP adresu serveru poskytovatele VoIP služeb a přesměrování paketů na tyto porty přicházejících na vnitřní IP adresu ústředny. Zároveň je nutno v konfiguračním souboru ústředny *sip.conf* nastavit hodnotu *externip* na sdílenou veřejnou IP adresu směrovače/firewallu. Z hlediska zabezpečení se nejedná o

vysoké riziko, neboť víme, že na těchto UDP portech nebude poslouchat žádná jiná služba a že na tyto porty bude přímo přistupovat pouze server poskytovatele. Požadavky ostatních zařízení v Internetu na tyto porty budou směrovačem/firewallem zahazovány.

V případě vyšších požadavků na bezpečnost lze umístit ústřednu do DMZ (demilitarizovaná zóna)⁶. V tom případě platí stejné pravidlo pro přesměrování UDP portů z Internetu jako v předchozím řešení, avšak na firewallu je navíc nutno povolit nejlépe veškerou příchozí UDP komunikaci z vnitřní sítě a odchozí UDP komunikaci do vnitřní sítě na portech, které jsou očekávány jednotlivými zařízeními. To může být v případě použití různých typů zařízení problematické, neboť různá zařízení mohou očekávat příchozí RTP komunikaci na různých rozsazích UDP portů. Například hardwarový telefon Grandstream GXP2000 sice umožňuje nastavit lokální RTP port, který se bude snažit s ústřednou vyjednat pro komunikaci, avšak toto nastavení není chápáno jako statický port nýbrž jako nejnižší možný počáteční port. Jiná zařízení mohou používat jiná nastavení a pravidla, která navíc nelze uživatelsky změnit. Zjištění potřebného rozsahu portů může být tedy složité.

Pro naše potřeby budeme dále kvůli zjednodušení uvažovat pouze první řešení, kdy je ústředna umístěna ve vnitřní síti a ne v DMZ.

⁶ [http://en.wikipedia.org/wiki/DMZ_\(computing\)](http://en.wikipedia.org/wiki/DMZ_(computing))

3.6 IAX2

IAX2 (Inter-Asterisk eXchange) ([1] – strana 619) je protokol vyvinutý formou Digium pro komunikaci mezi ústřednami, což však neznamená, že je limitován pouze pro tuto komunikaci. Je podporován i některými výrobci HW telefonů (např. Interbell) a některými SW telefony (např. Sflphone nebo Kfax). Příkazy a parametry jsou rozdílné od SIP posílány v binárním formátu. Hlavní výhodou IAX2 oproti SIP je to, že pro komunikaci používá jediný UDP port (4569) jak pro signalizaci, tak pro přenos multimediálních proudů. V praxi to znamená, že tento protokol snadno prochází všemi překlady adres (NAT). Mimo to je schopen spojit více současných relací (hovorů) do jednoho propojení (trunk), čímž se docílí snížení požadavku na přenosové pásmo a zároveň se sníží doba odezvy (zpoždění, latence).

Nevýhodou může být vyšší zátěž serveru a relativně malý počet hovorů zařaditelných do jednoho propojení. Jedná se o cca 300 – 400 současných hovorů [13], což však v případě malých až středních firem není příliš limitující. Další nevýhodou z toho vyplývající je to, že propojení pomocí IAX2 protokolu mnoho operátorů právě díky předchozímu omezení nepodporuje.

Ústřednu je možno v případě použití IAX2 protokolu pro komunikaci s ústřednou poskytovatele zapojit bez problémů jak do vnitřní sítě, tak do DMZ. U obvyklé konfigurace není nutno povolovat žádná příchozí spojení z Internetu. Pokud bude ústředna umístěna do DMZ, platí pro komunikaci mezi DMZ a vnitřní sítí stejná pravidla, jako v případě použití SIP.

4. Audio formát G.711

RTP i IAX2 používají pro přenos hlasu různé datové formáty. O převod analogového zvuku do požadovaného formátu se používají tzv. kodeky. Přestože existuje přibližně sedm audio formátů a s nimi souvisejících kodeků (některé z nich jsou licencované za poplatek), je nejpoužívanějším a v mnoha případech jediným audio formátem nabízeným našimi VoIP operátory formát G.711 ([1] - strana 626), který si zde krátce popíšeme.

G.711 [1], [14] je základním kodekem používaným v telekomunikacích (nejen v IP telefonii, ale i v klasické telefonii – PSTN⁷), ze kterého vycházejí i ostatní kodeky. Přenosová rychlost kodeku je 64 kb/s, k tomu je však nutno přičíst ještě režii cca 23 kb/s pro IP, UDP a RTP, takže celková požadovaná šířka pásma pro jeden směr přenosu je přibližně 87 kb/s. Výhodou kodeku G.711 je jak vysoká kvalita zvuku, tak velmi nízké nároky na využití CPU, nevýhodou pak vyšší požadavky na přenosovou rychlost. Další výhodou je, že je volně dostupný, neboť patenty na něj již vypršely⁸[14].

G.711 má dvě verze: μ -law (také označovaný jako U-law) a A-law. μ -law je používán převážně v severní Americe a Japonsku, A-law pak v Evropě a ostatních zemích, tedy i v České republice. A-law je speciálně vytvořen tak, aby jeho počítačové zpracování bylo co nejjednodušší.

Základní rozdíl mezi μ -law a A-law spočívá v tom, že μ -law používá vyšší kompresi z důvodu, že v severní Americe a Japonsku je v telefonních sítích pro přenos hlasu použito pouze 7 bitů a osmý bit je použit pro signalizaci. Evropa používá pro přenos hlasu všech 8 bitů a pro signalizaci se používá samostatný kanál. Díky tomu má tento přenos (přenášený hlas) vyšší kvalitu než v Americe a Japonsku.

⁷ Public switched telephone network – viz např.

http://en.wikipedia.org/wiki/Public_switched_telephone_network

⁸ http://www.itu.int/ITU-T/recommendations/related_ps.aspx?id_prod=911

5. Požadavky na hardware, operační systém a konektivitu

Požadavky se vždy odvíjejí od současného předpokládaného stupně využití systému a od toho, jaký se očekává další vývoj jeho využití (v praxi se samozřejmě většinou očekává zvýšení stupně využití). Přestože přesně popsat nároky jakéhokoliv systému je velmi náročné, ne-li takřka nemožné, lze alespoň navrhnout, podle jakých kritérií by měla být zvolena platforma, na které bude systém provozován.

5.1 Požadavky na hardware

Pro přibližné určení požadavků Asterisku na hardware (vycházíme z [1]) by neměl být měřítkem počet připojených uživatelů ale především počet současně otevřených komunikačních kanálů a typ používaných kodeků. Je nutno si uvědomit, že každý hovor otevře minimálně dva kanály nehledě na to, zda se jedná o hovor mezi jednotlivými zařízeními v rámci pobočky nebo o hovor mezi pobočkovým zařízením a zařízením ve veřejné telefonní síti nebo u VoIP operátora. Jeden kanál slouží pro komunikaci volajícího zařízení s místní ústřednou a druhý pro komunikaci mezi místní ústřednou a volaným zařízením (cizí ústředna, místní zařízení atd.).

Měla by být dodržena zásada, že pro zpracování hovorů (tedy nejdůležitější činnost ústředny) by měl být vyhrazen server, na kterém nebudou provozovány žádné další služby, jako jsou databázový server pro volitelné zaznamenávání historie hovorů nebo webový server obsluhující volitelné grafické rozhraní pro správu ústředny.

Na straně 640 zdroje [1] je uvedena tabulka zobrazující přibližné hardwarové požadavky. Tabulka uvedená na následující straně z ní vychází.

Tabulka 2 – vodítko pro určení požadavků na hardware

Využití	Počet kanálů	Minimální doporučená konfigurace
system na hraní	méně než 5	400-MHz x86, 256 MB RAM
malá nebo domácí kancelář	5 až 10 kanálů	1-GHz x86, 512 MB RAM
system pro malou firmu	10 až 25 kanálů	3-GHz x86, 1GB RAM
střední až velký system	více než 25 kanálů	Dvě a více CPU nebo vícejádrový CPU, případně distribuovaný system s více servery

Kromě počtu současně otevřených komunikačních kanálů a typu používaných kodeků je nutno brát ohled i na další aspekty. Například pořádání telekonference značně zatíží CPU, neboť je potřeba v reálném čase spojit ("mixovat") všechny příchozí zvukové proudy účastníků konference do jednoho a ten opět rozeslat všem účastníkům.

Dalším důležitým aspektem je logika vytáčeního plánu (způsob, jakým budou zpracovány příchozí a odchozí hovory). V případě, že jsou z vytáčeního plánu volány externí skripty nebo programy, dochází k dalšímu zatížení. Je vhodné co nejvíce logiky zahrnout do vytáčeního plánu a minimalizovat volání externích programů. Ty programy či skripty, které jsou volány, by měly být co nejvíce optimalizovány pro co nejmenší zátěž procesoru.

Co se týče kodeků, je nutno nalézt kompromis mezi potřebou co nejnižšího využití šířky přenosového pásma a potřebou co nejnižšího vytížení CPU. V našem konkrétním případě si však s tímto aspektem nemusíme dělat starosti, neboť díky tomu, že většina našich operátorů podporuje pro připojení firemní telefonní ústředny obvykle jen kodek G.711 A-law, budeme používat pouze tento. Ačkoli má nejvyšší požadavky na šířku přenosového pásma, požadavky na využití CPU má nejnižší.

Nakonec tedy, jako i ve většině jiných případů, platí to, že před nasazením ústředny do ostrého provozu je nutno provést testování, které teprve potvrdí správnou volbu hardware.

V případě bližšího zájmu je možno prohlédnout si stránku

<http://www.voip-info.org/wiki/view/Asterisk+dimensioning>

věnující se problematice určení požadavků Asterisku na hardware.

Pro účely této práce si vzhledem k očekávanému nízkému vytížení systému během testovacího provozu dovolíme použít virtuální počítač. Jako virtualizační prostředí může být použit například Hyper-V server firmy Microsoft nebo Oracle VirtualBox⁹ (dále jen VirtualBox) distribuovaný pod GNU/GPL¹⁰.

5.2 Požadavky na operační systém

Asterisk by měl jít zkompilovat, nainstalovat a provozovat na jakékoliv distribuci Linuxu. Zdroj [1] se zmiňuje především o distribucích CentOS (strana 35 zdroje) a Ubuntu (strana 40 zdroje). Asterisk je však podporován i UNIXovými systémy, jako je např. již v kapitole 2 zmíněné FreeBSD, jež pro tuto práci použijeme. FreeBSD je zvoleno z důvodu jeho stability a z důvodu osobní preference a větších zkušeností autora před Linuxem. Nehledě na výběr OS však bude konfigurace Asterisku na jakémkoliv z nich shodná.

Ať zvolíme jakýkoliv operační systém (OS), je důležité, aby byl systém nainstalován v co nejmenší možné konfiguraci a především bez grafického rozhraní zatěžujícího systém. Pro správnou funkci ústředny (například činnost hlasové schránky nebo fronty volajících či jiné činnosti/funkce závislé na přesném čase) je také nutné mít nastaven správný systémový čas. Proto je důležité správně nastavit synchronizaci času s libovolným NTP (Network Time Protocol)¹¹ serverem (u nás lze použít např. server ntp.karpo.cz).

⁹ <https://www.virtualbox.org/>

¹⁰ <http://www.gnu.org/licenses/gpl.html>

¹¹ http://cs.wikipedia.org/wiki/Network_Time_Protocol

5.3 Požadavky na konektivitu

Požadavky na šířku pásma lze snadno spočítat na základě přenosových rychlostí jednotlivých kodeků. Stačí vynásobit přenosovou rychlost použitého kodeku včetně režie požadovaným počtem současných kanálů. V případě námi použitého kodeku G.711 A-law tedy bude výsledná šířka pásma rovna 87 kb/s vynásobeným počtem současných kanálů a to jak v dopředném tak zpětném směru. Pro 20 současných hovorů (malá firma) dostaneme požadovanou rychlost $87 * 20 \text{ kb/s} = 1740 \text{ kb/s}$, což je rychlost, kterou jsou schopni poskytnout snad všichni naši internetoví operátoři (samozřejmě nebereme ohled na vytáčené připojení, které je pro IP telefonii naprosto nepoužitelné).

Rychlost však není jediným kritériem, velmi důležitá je také kvalita připojení. Kromě ztrátovosti paketů, která by měla být maximálně 1-5% (nejlépe však žádná), je důležitá i doba trvání cesty hlasových dat z jednoho zařízení na druhé. Literatura ([1] – strana 629) uvádí, že lze obecně akceptovat pravidlo, že když jsme schopni doručit zvuk k posluchači do 150 ms, je možný normální tok rozhovoru. V praxi to pro nás znamená, že doba odezvy připojení (doba mezi odesláním požadavku vzdálenému zařízení a mezi přijetím jeho odpovědi na tento požadavek) mezi dvěma zařízeními (telefony) by měla dosahovat maximálně 300ms. Tato hodnota však platí pro celkovou dobu průchodu signálu všemi dalšími zařízeními, jež jsou mezi jednotlivými telefony. Bude-li mezi naší ústřednou a naším VoIP operátorem odezva 300ms, ještě to neznámá, že bude hovor vyhovující. Je nutno přičíst dobu odezvy mezi naším operátorem a operátorem volaného (předpokládejme však, že ta je minimální a to v jednotkách milisekund) a k tomu dobu odezvy mezi volaným a jeho operátorem. V případě, že by se všichni uživatelé drželi pravidla, že jejich připojení musí splňovat podmínku maximálně 300ms odezvy, byla by v extrémním případě celková doba doručení zvuku něco málo přes 600ms, což by mohlo vést k nesrozumitelné a frustrující komunikaci. Měli bychom se tedy řídit pravidlem, že maximální průměrná doba odezvy mezi ústřednou a VoIP operátorem by neměla přesáhnout 75ms. Nejlépe by však měla být maximálně cca 60ms.

Podle měření pomocí příkazu ping se v síti českého UPC doba odezvy mezi směrovačem a výchozí bránou poskytovatele pohybuje průměrně kolem 13ms, v optické síti T-Systems a

bezdrátové síti Coprosysu pak kolem 1ms. Ztrátovost paketů mezi směrovačem a výchozí branou se jak v síti UPC tak v i dalších zmíněných sítích pohybuje v ideálním případě mezi 0-1%. Na základě těchto zkušeností lze předpokládat, že i ostatní větší poskytovatelé připojení k Internetu dosahují srovnatelných parametrů. Ideální stav je ten, kdy telefonní ústředna VoIP poskytovatele je umístěna přímo v síti používaného poskytovatele připojení k Internetu, neboť v takovém případě se lze spolehnout na minimální dobu odezvy.

V případě, že je Internetové připojení používáno i k jiným účelům než k provozování IP telefonie, což je obvyklý stav, je vhodné nastavit pro komunikaci mezi ústřednou a poskytovatelem VoIP nejvyšší prioritu (umožňuje-li to směrovač), aby nedocházelo k nečekaným a nepříjemným výpadkům zvuku (hlasu).

6. Vlastní návrh řešení

6.1 Instalace

V případě instalace FreeBSD a ústředny Asterisk na fyzický počítač s námi zvolenými parametry můžeme přejít přímo na kapitolu 6.1.2. Popis základní instalace FreeBSD 9.1 pro uživatele, kteří nemají s tímto systémem zkušenosti, je uveden v textové příloze číslo 1. Pro instalaci použijeme ISO obraz disku umístěný na Internetové adrese

<http://ftp.cz.freebsd.org/pub/FreeBSD/ISO-IMAGES-i386/9.1/FreeBSD-9.1-RELEASE-i386-disc1.iso>,

který je v případě instalace na fyzický počítač nutno nejprve vypálit na CD-ROM. Tento ISO obraz je zároveň součástí elektronické přílohy č. 3.

Všechny níže uvedené změny v konfiguračních souborech systému i ústředny lze pod systémem FreeBSD provést pomocí vestavěných editorů `vi` nebo `ee`. Uživatelé, kteří s těmito editory pracovat neumí, nebo nechtějí, mohou použít například editor programu Midnight Commander¹².

6.1.1 Příprava virtualizačního prostředí pro vzorovou instalaci

Pro vzorovou instalaci ústředny je použit virtuální počítač s 2 x Intel Xeon CPU a 2GB RAM. Jako virtualizační prostředí je pro účely této práce zvolen Hyper-V server a VirtualBox, v praxi však lze použít libovolné virtualizační prostředí, na kterém bude systém FreeBSD schopen síťového provozu. Virtuální disk ve formátu VHD¹³, jehož velikost je nastavena na 15GB, lze použít jak pod VirtualBoxem tak pod Hyper-V. Uvedený VHD Disk se vzorovou instalací je součástí elektronické přílohy č. 4.

Budeme-li přenášet tento již nainstalovaný virtuální počítač mezi Hyper-V a VirtualBoxem, nesmíme zapomenout změnit v souboru `/etc/rc.conf` (je součástí textové přílohy č. 4 a elektronické přílohy č. 2) název konfigurace síťového adaptéru. Pod

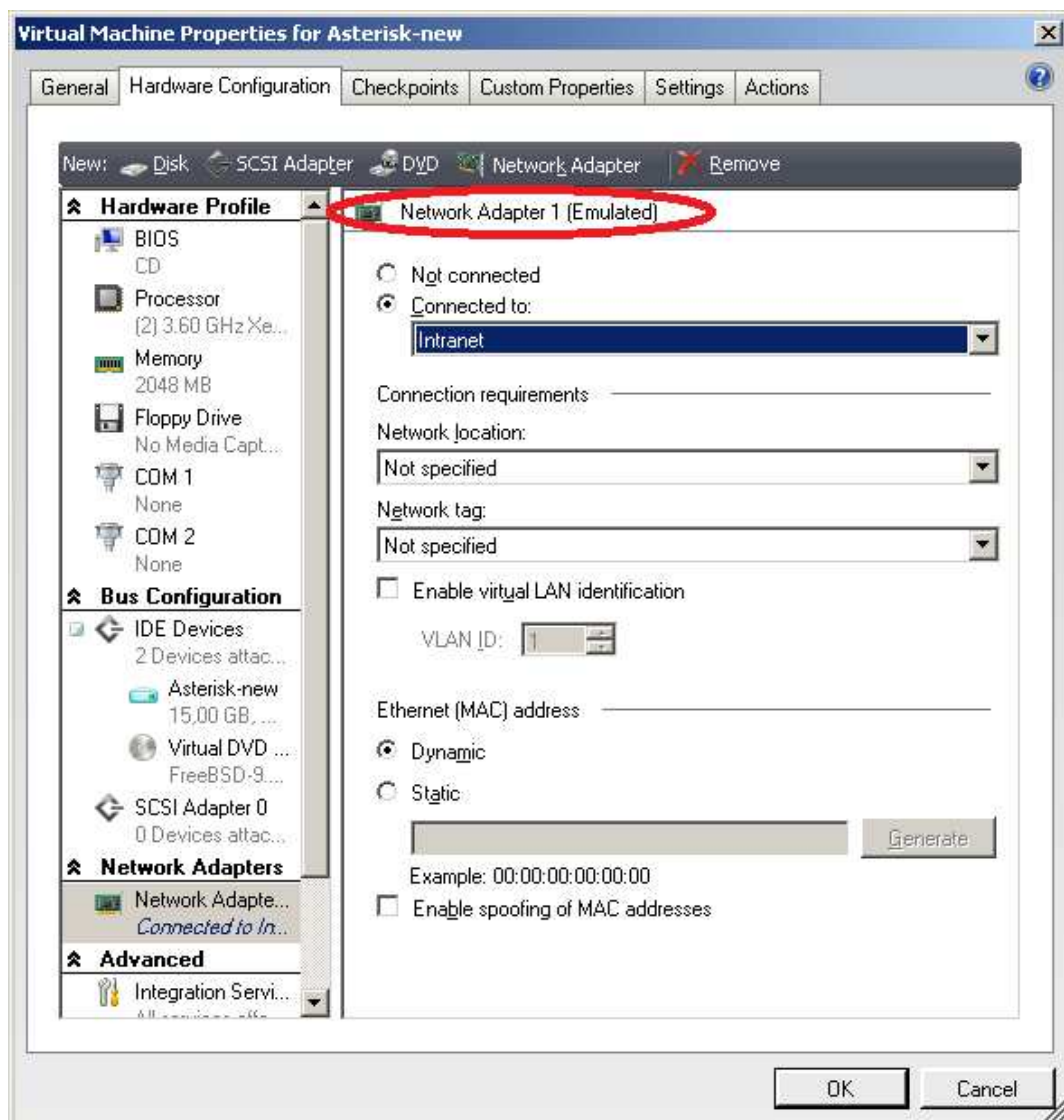
¹² <http://www.midnight-commander.org/>

¹³ [http://en.wikipedia.org/wiki/VHD_\(file_format\)](http://en.wikipedia.org/wiki/VHD_(file_format))

VirtualBoxem je konfigurace pojmenována jako `ifconfig_em0`, pod Hyper-V pak jako `ifconfig_de0`.

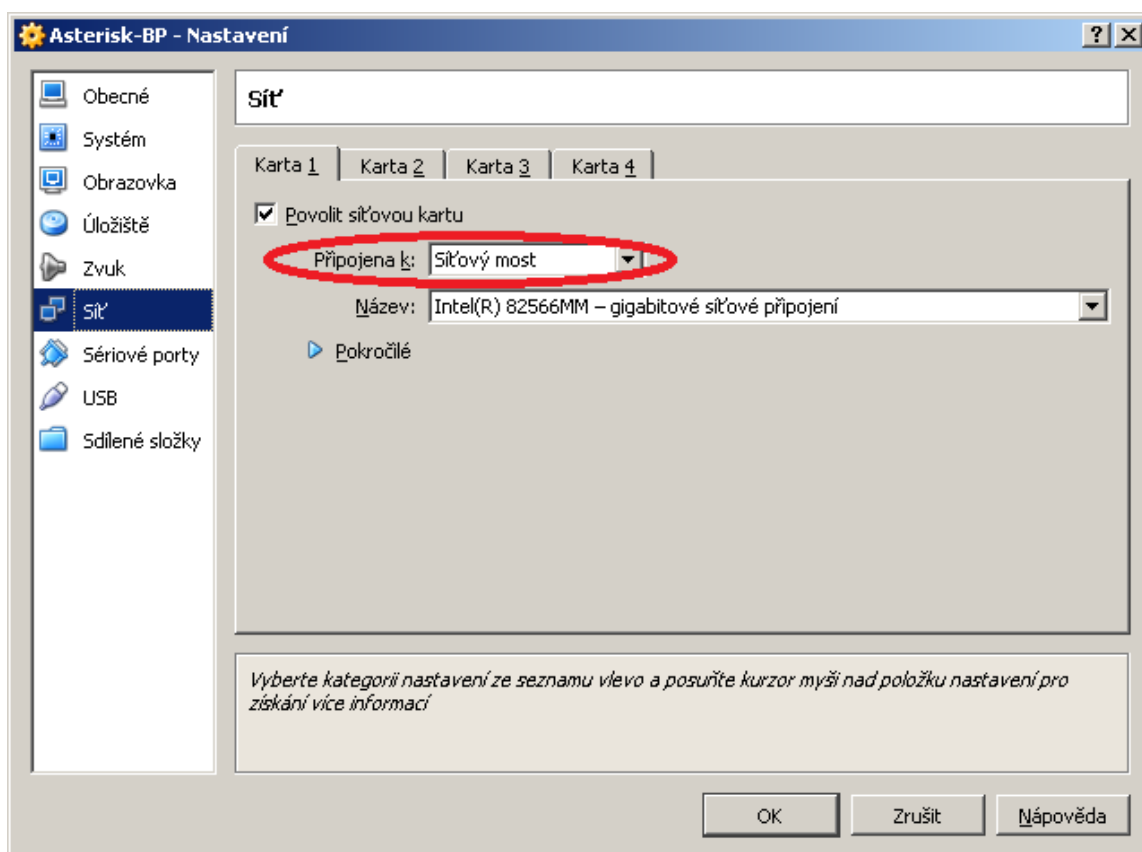
Před instalací FreeBSD na Hyper-V nebo před připojením VHD disku z přílohy je nutno ve vlastnostech virtuálního počítače nastavit síťový adaptér jako **emulovaný** (FreeBSD nedokáže rozpoznat výchozí syntetický síťový adaptér nabízený Hyper-V) a připojit jej do vnitřní sítě.

Obrázek 1 - nastavení síťového adaptéru pod Hyper-V



VirtualBox (v době psaní této práce ve verzi 4.2.6) již má předdefinovaná nastavení, mezi nimiž je obsaženo i funkční nastavení pro systém FreeBSD. V nastavení je nutno před instalací nebo připojením VHD disku změnit pouze typ síťového připojení na **Síťový most** (viz Obrázek 2 – *Nastavení síťového adaptéru pod VirtualBoxem*).

Obrázek 2 – *Nastavení síťového adaptéru pod VirtualBoxem*



6.1.2 Instalace telefonní ústředny Asterisk

Máme-li nainstalovaný a nakonfigurovaný operační systém (v našem případě FreeBSD), můžeme přistoupit k samotné instalaci telefonní ústředny. Vzhledem k tomu, že jediná zásadní konfigurace probíhá v prvním dialogu, není nutno uvádět jednotlivé obrázky zobrazovaných konfiguračních dialogů.

Samotný software ústředny nainstalujeme z kolekce portů pomocí příkazu

```
cd /usr/ports/net/asterisk; make install clean
```

V prvním konfiguračním dialogu necháme zapnuté pouze volby CURL, FREETDS (podpora přístupu k databázím Sybase a MS SQL), GSM (podpora kodeku GSM), ODBC (podpora přístupu k databázím pomocí ODBC), SPANDSP (podpora faxování), VORBIS (podpora kodeku Ogg Vorbis) a UNIXODBC.

V dalších konfiguračních dialogích, které se zobrazují postupně během instalace součástí požadovaných pro chod ústředny, ponecháme všechny volby v jejich výchozím nastavení kromě čtvrtého (konfigurace FreeTDS), kde zvolíme pouze UNIXODBC.

Celková doba kompilace a instalace ústředny včetně požadovaných součástí může trvat v závislosti na výkonu počítače přibližně 20 až 40 minut.

6.2 Nastavení telefonní ústředny

Nyní máme vše připraveno pro zprovoznění telefonní ústředny a můžeme začít s jejím nastavením. Nebude-li zmíněno jinak, budou všechny konfigurační soubory umístěny v adresáři `/usr/local/etc/asterisk`. Tato kapitola vychází kromě literatury [1] i z [15], [16] a vlastních zkušeností a experimentů. Popis některých nastavení a funkcí, které jsou zdokumentovány nepřesně či nedostatečně, bylo nutno dohledat pomocí vyhledávače v různých internetových diskusích týkajících se ústředny Asterisk (tyto diskuse nejsou vzhledem k jejich množství a kusým informacím uvedeny v seznamu zdrojů).

Pro automatické spouštění ústředny po startu systému je nutno do souboru `/etc/rc.conf` přidat následující řádky:

```
asterisk_enable="YES"
asterisk_flags="-p"
```

Chceme-li zajistit, aby se služba ústředny po každém jejím plánovaném ale především neplánovaném vypnutí automaticky spustila, vytvoříme hlídací skript (tzv. Watchdog), který se o spuštění ústředny postará. Skript pojmenovaný například `astdog.sh` (je součástí elektronické přílohy č. 2) bude vypadat takto:

```
#!/bin/sh
sleep 300
while [ 1 ]; do
    BEZI=`ps -fax|grep 'asterisk -p'|grep -v grep|wc -l`;
    echo $BEZI
    if [ $BEZI = 0 ]; then
        /usr/local/sbin/asterisk -p;
    fi
    sleep 1
done
```

Skript umístíme do nově vytvořeného adresáře `/usr/local/etc/asterisk/bin` a pomocí příkazů

```
chown root /usr/local/etc/asterisk/bin/astdog.sh
chmod 544 /usr/local/etc/asterisk/bin/astdog.sh
```

nastavíme oprávnění k jeho spuštění uživatelem `root`.

Automatické spuštění skriptu po startu systému zajistíme přidáním souboru nazvaným například `astdog_rc.sh` do adresáře `/usr/local/etc/rc.d`. Obsah souboru `astdog_rc.sh` (je součástí elektronické přílohy č. 2) bude následující

```
#!/bin/sh
/usr/local/etc/asterisk/bin/astdog.sh&
```

Oprávnění k jeho spuštění uživatelem `root` nastavíme pomocí příkazů

```
chown root /usr/local/etc/rc.d/astdog_rc.sh
chmod 544 /usr/local/etc/rc.d/astdog_rc.sh
```

6.2.1 Nastavení modulů zaváděných při startu

Asterisk je modulární systém a lze nastavit, které moduly se mají při startu zavádět. Ve výchozím nastavení jsou pomocí direktivy `autoload=yes` zavedeny všechny moduly z adresáře `/usr/local/lib/asterisk/modules`.

Změna nastavení není bezpodmínečně nutná, neboť zavedení modulů je rychlé a nejsou-li využívány, mají minimální režii. Budeme-li chtít v budoucnu některé z modulů zakázat, provedeme toto nastavení v souboru `modules.conf` (součást elektronické přílohy č. 2 a textové přílohy č. 3) do kterého přidáme pro každý modul, který nechceme zavádět řádek ve formátu

```
noload => jmeno_modulu.so
```

Je tedy pouze na naší volbě, zda ponecháme soubor `modules.conf` v jeho výchozím nastavení, nebo zakážeme moduly uvedené v příloze. Jejich popis je pro naše účely

nepodstatný, pro více informací je možno nahlédnout na stranu 10 a následující strany literatury [1] případně dohledat informace na Internetu.

6.2.2 Nastavení protokolu SIP a koncových zařízení – sip.conf

V souboru `/usr/local/etc/asterisk/sip.conf` nastavujeme nejen vlastnosti protokolu SIP, ale také definujeme seznam zařízení, která mohou ústřednu používat, případně seznam poskytovatelů, ke kterým se ústředna připojuje.

Do souboru lze psát komentáře, jež se uvozují znakem `;` (středník) a mohou se vyskytovat kdekoliv v textu. Kompletní soubor `sip.conf` je uveden v textové příloze č. 5 a v elektronické příloze č. 2.

Základní nastavení protokolu je definováno v sekci `[general]`. Většinu globálních nastavení obvykle nedefinujeme a jejich výchozí nastavení ponecháme na ústředně. Pro účely vzorové konfigurace budeme sekci globálních nastavení definovat následujícím způsobem:

```
[general]
context=bez-registrace ; Vychodi kontext
allowguest=yes ; Povoleni hovoru z neregistrovaniych zarizeni
udpbindaddr=0.0.0.0 ; Ustredna bude naslouchat na libovolnem
                        ; sitovem rozhrani
disallow=all ; Zakazeme vsechny kodeky
allow=alaw ; Povolime pouze kodek alaw
```

Proměnná `context` obsahuje název sekce v souboru `extensions.conf` (nastavení vytáčecího plánu), ve které je definováno, jakým způsobem se mají hovory (příchozí i odchozí) daného zařízení zpracovat. Termín **kontext** zde tedy můžeme chápat jako pojítko mezi definicí zařízení a pojmenovanou částí vytáčecího plánu, nebo lépe jako vstupní bod vytáčecího plánu.

Kontext `bez-registrace` je přiřazen všem neregistrovaným zařízením (zařízení, pro které není v souboru `sip.conf` uvedena definice).

Proměnná `allowguest` určuje, zda budou povolena přímá volání pomocí protokolu SIP neregistrovaným zařízením. Pokud je nastavena na `yes`, budou takováto volání zpracována pomocí vytáčekého plánu pro kontext `bez-registrace`. Je-li nastavena na `no`, bude hovor ihned odmítnut bez zpracování vytáčekým plánem.

Přímé volání pomocí SIP spočívá v zadání URI (Uniform Resource Identifier - jednotný identifikátor zdroje) ve formátu

```
SIP:identifikator.zarizeni@adresa_serveru
```

do telefonu podporujícího tento způsob volání. Například po zadání URI

```
SIP:1111@asterisk.czu.cz
```

bychom byli v případě existence takového serveru spojeni s vnitřní linkou 1111.

Základní výhodou takového volání je, že při správném nastavení vytáčekého plánu se lze dovolat na čísla vnitřních linek bez jakýchkoliv poplatků telekomunikačním operátorům. Mezi softwarové telefony podporující tato volání patří volně šiřitelný **Phoner**¹⁴, jenž je součástí elektronické přílohy č. 5.

Proměnná `udpbindaddr` určuje, na jakých síťových rozhraních bude ústředna naslouchat požadavkům zaslaným pomocí UDP. Lze nastavit buď jedinou IP adresu (ústředna bude naslouchat pouze na rozhraní s touto adresou) nebo hodnotu `0.0.0.0`, která určuje, že ústředna bude naslouchat na všech rozhraních. Pozor, v případě, že máme tři síťová rozhraní, nelze z nich vybrat dvě, ale buďto pouze jedno, nebo všechna.

Proměnné `disallow` a `allow` slouží k určení kodeků použitých pro přenos hlasových dat. Nastavením `disallow=all` zakážeme všechny dostupné kodeky a pomocí proměnné `allow` určíme pouze ty, které chceme používat. Proměnná `allow` se může opakovat pro libovolný počet kodeků, to znamená, že do sekce `[general]` bychom pro povolení kodeků `alaw` a `ulaw` zapsali tyto řádky:

```
disallow=all
allow=alaw
allow=ulaw
```

¹⁴ <http://www.phoner.de/>

Definice jednotlivých zařízení je z důvodu přehlednosti vhodné uvádět ve zvláštním souboru (nebo i více souborech), který lze do výchozího konfiguračního souboru zahrnout direktivou `#include`. Řekněme, že ústředna je konfigurována pro firmu zabývající se vývojem software (typ firmy však není podstatný, princip bude stejný pro libovolnou firmu). Operátor nám přidělil provolbu 100 čísel (např. 123 123 100 až 123 123 199 – vnitřním linkám tedy můžeme přiřadit dvouciferná čísla 00 až 99) a telefonní čísla chceme rozdělit na čtyři skupiny podle jednotlivých oddělení (obchod, vývoj, technická podpora a ostatní). Pro každou ze skupin bude vytáčecí plán odlišný, zařízení v jednotlivých skupinách tedy budou mít odlišné kontexty. Čísla budou rozdělena například takto:

Podpora: 123 123 100 – 123 123 119

Obchod: 123 123 140 – 123 123 159

Vývoj: 123 123 160 – 123 123 179

Ostatní: 123 123 180 – 123 123 190

Nezařazená čísla: 123 123 120 – 123 123 139 a 123 123 191 – 123 123 199.

Vytvoříme pět souborů s názvy `sip_operator.conf`, `sip_obchod.conf`, `sip_vyvoj.conf`, `sip_podpora.conf` a `sip_ostatni.conf`. Tyto soubory zahrneme do souboru `sip.conf` pomocí direktivy `#include` takto:

```
#include sip_operator.conf
#include sip_obchod.conf
#include sip_vyvoj.conf
#include sip_podpora.conf
#include sip_ostatni.conf
```

Dále vytvoříme soubor `sip_sablony.conf` a v něm globální šablonu

[`sablona-vse`] společnou všem vnitřním linkám/zařízením a šablony společné jednotlivým skupinám linek/zařízení ([`sablona-vyvoj`], [`sablona-obchod`], [`sablona-podpora`] a [`sablona-ostatni`]), které budou dědit nastavení šablony [`sablona-vse`] (Systém šablon nám usnadní vytváření i budoucí změny definic zařízení.). Definice každé šablony je odlišena od definice zařízení tím, že za jejím názvem je uveden řetězec (`![,nazev_nadrazene_sablony]`).

Podle těchto šablon pak již vytvoříme jednotlivé definice koncových zařízení. Soubor musíme zahrnout do souboru `sip.conf` direktivou `#include` na úplný začátek souboru. Pozor, je-li zahrnut za sekci `[general]` (tedy ne na začátku), nefunguje z neznámého důvodu propojení s operátorem. Toto je ověřeno experimentálně.

Kompletní soubor `sip_sablony.conf` (součást elektronické přílohy č. [2]) bude vypadat takto:

```
[sablona-vse](!)
type=friend
secret=456
host=dynamic
qualify=yes
dtmfmode=rfc2833
deny=0.0.0.0/0.0.0.0
permit=192.168.0.0/255.255.0.0

[sablona-vyvoj](!,sablona-vse)
context=vyvoj
callgroup=1
pickupgroup=1

[sablona-obchod](!,sablona-vse)
context=obchod
callgroup=2
pickupgroup=2

[sablona-podpora](!,sablona-vse)
context=podpora
callgroup=3
pickupgroup=3,2

[sablona-ostatni](!,sablona-vse)
context=ostatni
callgroup=4
pickupgroup=4
```

Řetězec (!, *sablona-vse*) za názvem šablony (např. [*sablona-vyvoj*]) tedy znamená, že šablona je opravdu šablonou (znak !) a že dědí nastavení šablony *sablona-vse*.

Proměnná *type* určuje, jakým způsobem bude zařízení ústřednou obsluhováno. Může nabývat hodnot *user*, *peer* nebo *friend*. Následující tři odstavce vycházejí z [17] a [1].

Hodnota *peer* znamená, že zařízení je registrováno pro příchozí hovory. Asterisk si udržuje ve své interní tabulce *peerů* informaci o jménu, IP adrese zařízení a jeho portu. Seznam *peerů* lze zobrazit pomocí příkazu *sip show peers* v CLI (Command Line Interface – česky příkazový řádek) ústředny. Díky tomu existuje způsob, jak zařízení kontaktovat. V tabulce uživatelů pak o tomto zařízení neudrží žádné informace. Toto zařízení může hovory jak přijímat, tak iniciovat.

Hodnota *user* znamená, že zařízení se neregistruje pro příchozí hovory, ale je schopno provádět odchozí hovory. Není definován způsob, jak zařízení zavolat, neboť Asterisk si v tabulce *peerů* neudrží o tomto zařízení žádné informace (ty má pouze v tabulce uživatelů, jejichž seznam lze zobrazit pomocí příkazu *sip show users* v CLI ústředny) a tedy neví, jak je kontaktovat. Jednoduše řečeno, *user* je schopen iniciovat volání, ale nikdo se mu nedovolá.

Hodnota *friend* je kombinací obou předchozích pravidel. To znamená, že Asterisk si udržuje informace o zařízení jak v seznamu *peerů*, tak v seznamu uživatelů. Pro SIP telefony je nejpoužívanější a budeme ji tedy pro zjednodušení používat i my.

Proměnná *secret* určuje heslo, které je nutno zadat na straně klienta při registraci k ústředně. Pro zjednodušení budou mít všechna naše zařízení stejné heslo (456).

Proměnná *host* určuje IP adresu daného zařízení. V případě koncového VoIP zařízení nastavujeme proměnnou obvykle na hodnotu *dynamic*, neboť jeho IP adresa se může měnit. V případě operátora nastavujeme jeho pevnou IP adresu.

Proměnná `qualify` říká ústředně, zda má monitorovat stav zařízení. Je-li nastavena na `yes`, je v seznamu `peerů` (zobrazeném pomocí příkazu `sip show peers`) zobrazen stav zařízení a v případě, že je připojen a stav má hodnotu `OK`, je též zobrazena doba odezvy komunikace mezi zařízením a ústřednou. Je-li nastavena na `no`, je místo stavu zobrazena hodnota `Unmonitored`.

Proměnná `dtmfmode` určuje způsob přenosu DTMF tónů mezi zařízením a ústřednou. Její hodnota by měla být nastavena podle způsobu přenosu DTMF nastaveného na telefonu. Obvykle se používá hodnota `rfc2833`, což je i výchozí hodnota. Ve vzorové konfiguraci linek tedy bude tato proměnná uvedena pouze v souboru `sip_operator.conf`.

Proměnná `deny` určuje, ze kterých IP adres se nesmí definované zařízení připojit. V našem případě `deny=0.0.0.0/0.0.0.0` znamená, že definované zařízení se nesmí připojit nikud. Připojení povolíme až pomocí proměnné `permit`. Nastavení naší proměnné `permit` na hodnotu `192.168.0.0/255.255.0.0` znamená, že zařízení může ústřednu použít pouze v případě, že jeho IP adresa je z rozsahu adres `192.168.0.1` až `192.168.255.254`. Řídíme se tedy pravidlem, že nejprve je potřeba veškerý přístup zakázat a explicitně povolit přístup pouze vybraným IP adresám.

Proměnná `context` definuje název vstupního bodu vytáčeného plánu - kontext v souboru `extensions.conf`. To znamená, že všechny příchozí i odchozí hovory budou zpracovány podle pravidel v této sekci uvedených.

Číselná proměnná `callgroup` slouží k zařazení jednotlivých zařízení do různých skupin příjemců hovorů. Zařízení může být členem pouze jedné skupiny. Proměnná `pickupgroup` pak uvádí seznam skupin oddělených čárkou, jejichž příchozí hovory může uživatel daného zařízení převzít pomocí zadání kombinace `*8` na číselníku telefonu.

Mějme například tři zařízení, s rozdílnými hodnotami proměnné `callgroup`.

```
[01]
callgroup=1
pickupgroup=1,3
[02]
callgroup=2
pickupgroup=2,3
[03]
callgroup=3
pickupgroup=1-3
```

Zařízení 01 pak bude schopno převzít hovor přicházející na zařízení 03 a všechna další zařízení, jejichž `callgroup` je rovna 1 nebo 3.

Zařízení 02 bude schopno převzít hovor přicházející na zařízení 03 a všechna další zařízení, jejichž `callgroup` je rovna 2 nebo 3.

Zařízení 03 bude schopno převzít hovor přicházející na zařízení 01 a 03 a všechna další zařízení, jejichž `callgroup` je mezi 1 a 3.

Pozor, tento způsob přebírání hovorů funguje pouze v rámci jednoho protokolu! To znamená, že je-li některý telefon ve skupině připojen pomocí jiného protokolu než SIP (např. IAX2), nelze příchozí hovor na toto zařízení převzít.

Definice jednotlivých zařízení potom již jednoduše definujeme pouze na základě výše uvedených šablon.

Soubor `sip_vyvoj.conf` bude vypadat takto:

```
[60](sablon-vyvoj)
[61](sablon-vyvoj)
...
...
[79](sablon-vyvoj)
```

Číslo v hranatých závorkách určuje jméno zařízení použité ve vytáčecím plánu a zároveň přihlašovací jméno uživatele. Pro zjednodušení zpracování hovorů budeme pojmenovávat jednotlivá zařízení podle koncových čísel provolby, čímž docílíme toho, že číslo linky bude shodné s názvem zařízení. V našem případě se jedná o poslední dvojčíslí. V případě, že by nám byla přidělena provolba 1000 čísel, použili bychom pro jméno poslední trojčíslí, v případě přidělené provolby 10 čísel, pak pouze poslední číslici. Text v kulatých závorkách (`sablona-vyvoj`) je odkaz na nadřazenou šablonu, jejíž nastavení zařízení dědí.

Soubor `sip_obchod.conf` bude vypadat takto:

```
[40](sablona-obchod)
[41](sablona-obchod)
...
...
[59](sablona-obchod)
```

Soubor `sip_podpora.conf` bude vypadat takto:

```
[01](sablona-podpora)
[02](sablona-podpora)
...
...
[19](sablona-podpora)
```

A nakonec soubor `sip_ostatni.conf` bude vypadat takto:

```
[80](sablona-ostatni)
[81](sablona-ostatni)
...
...
[90](sablona-ostatni)
```

Kompletní soubory `sip_*.conf` jsou součástí elektronické přílohy č. 2.

Máme-li nadefinovány vnitřní linky/zařízení, můžeme přistoupit k definici připojení ústředny k operátorovi.

Definice bude uložena v souboru `sip_operator.conf` (součást elektronické přílohy č. 2):

```
register => UZIVATEL:HESLO@@ 203.0.113.113
[Operator]
host=203.0.113.113
defaultuser=UZIVATEL
secret=HESLO
dtmfmode=inband
insecure=port,invite
canreinvite=no
context=prichozi
type=friend
disallow=all
allow=alaw
qualify=yes
```

Pro pojmenování vzdálené ústředny zvolíme hodnotu `Operator` (můžeme však zvolit libovolnou jinou hodnotu). Tento název budeme používat ve vytáčecím plánu pro iniciaci odchozích hovorů.

Proměnná `host` identifikuje v tomto případě IP adresu VoIP poskytovatele.

Proměnná `defaultuser` obsahuje přihlašovací jméno uživatele k operátorovi, proměnná `secret` pak operátorem přidělené heslo.

Proměnné `dtmfmode` je potřeba věnovat zvláštní pozornost, neboť nebude-li správně nastavena, nebude funkční přenos DTMF tónů v případě volání na různé audioslužby, jako je například automatická linka O2 (číslo 800 123 456). U operátorů Coprosys a T-Systems je funkční pouze hodnota `inband`, u ostatních operátorů však může být hodnota jiná. Tuto hodnotu je tedy potřeba vždy otestovat pro konkrétního operátora. Proměnná může nabývat

hodnot `inband`, `rfc2833`, `info` a `auto`. Je-li nastavena na `auto`, bude použita hodnota `rfc2833`, je-li druhou stranou nabídnuta, jinak se použije hodnota `inband`.

Proměnná `insecure` souvisí s autorizací požadavků vzdálené strany. Je-li nastavena na hodnotu `port`, nezáleží na tom, na jakém portu požadavek přišel. Je-li nastavena na `invite`, nejsou při požadavku vyžadovány přihlašovací údaje. Hodnota `port`, `invite` je kombinací obou předchozích. Tato hodnota se obvykle používá pro povolení příchozích hovorů, což je i náš případ.

Nastavení proměnné `canreinvite` na `no` zajistí, že hlasová data všech hovorů půjdou přes ústřednu a nebude prováděn pokus o přímé RTP spojení mezi koncovým zařízením a ústřednou operátora.

Pro proměnné `context`, `type`, `disallow`, `allow` a `qualify` platí již dříve uvedená pravidla (viz předchozí strany).

6.2.3 Nastavení hlasové schránky – voicemail.conf

Telefonní číslo 123 123 100 je vybráno jako obecné číslo technické podpory. Číslo 123 123 190 pak jako obecné číslo firmy (nejprve bude vyzvánět na recepci a postupně pak na linkách obchodního oddělení). Mimo pracovní dobu chceme dát zákazníkům možnost zanechat hlasovou zprávu.

Dříve než ve vytáčecím plánu nastavíme přechod do hlasové schránky, musíme ji nastavit v souboru `voicemail.conf` (součást elektronické přílohy č. 2). Konfigurace není složitá. Pro výše uvedená čísla (resp. jejich dvě koncové číslice) stačí přidat do sekce `[default]` konfiguračního souboru následující řádky (ostatní nastavení v souboru nebudeme měnit):

```
[default]
90 => ,90,schranka@firma.tld
00 => ,00,schranka@firma.tld
```

formát řádku je následující:

```
číslo => heslo[,jméno[,e-mail[,pager[,možnosti[|možnosti]]]]]
```

To znamená, že naše konfigurace obsahuje pouze číslo hlasové schránky, prázdné heslo (nebude vyžadováno), jméno schránky a e-mailovou adresu, na kterou budou hlasové zprávy přeposílány. E-mailovou adresu z příkladu je nutno nahradit platnou adresou. Předpokladem bezproblémového odesílání hlasových zpráv je správné nastavení názvových serverů v konfiguračním souboru systému `/etc/resolv.conf`.

6.2.4 Nastavení fronty volajících – queues.conf

Vzhledem k tomu, že na linku technické podpory může volat více zákazníků, než je konzultantů, je potřeba volající zařadit do fronty, ze které potom budou konzultanti jednotlivé hovory postupně vyzvedávat. Fronty volajících se nastavují v souboru `queues.conf`.

Na konec souboru queues.conf (součást elektronické přílohy č. 2) přidáme následující řádky (zbytek souboru ponecháme tak, jak je):

```
[DynamickaFrontaPodpory]
strategy=rrmemory
timeout=15
retry=3
periodic-announce-frequency=30
periodic-announce=podpora-2
joinempty=yes
leavewhenempty=yes

[StatickaFrontaPodpory]
strategy=ringall
timeout=15
retry=1
periodic-announce-frequency=30
periodic-announce=podpora-2
joinempty=yes
leavewhenempty=no
member => SIP/40
member => SIP/41
```

Máme tedy vytvořeny dvě fronty, jednu dynamickou (`DynamickaFrontaPodpory`) a další statickou (`StatickaFrontaPodpory`). Do dynamické fronty se budou operátoři přihlašovat zadáním čísla 96 a odhlašovat se z ní budou zadáním čísla 97. Ve statické frontě jsou linky, na které budou příchozí hovory směrovány, uvedeny napevno.

Proměnná `strategy` určuje, jakým způsobem bude probíhat vyzvánění na telefonech konzultantů. Hodnota `rrmemory` znamená, že hovory ve frontě postupně kolují po jednotlivých konzultantech v pořadí, v jakém se do fronty přihlásili s tím, že se v paměti udržuje číslo konzultanta, který převzal poslední hovor. Při příchodu dalšího hovoru bude předávání pokračovat od následujícího konzultanta a to opět v pořadí přihlášení do fronty. Hodnota `ringall` znamená, že při příchodu hovoru do fronty budou vyzvánět všechny telefony, jejichž linky jsou v ní uvedeny, zároveň.

Proměnná `timeout` určuje, kolik sekund bude trvat vyzvánění na telefonech jednotlivých konzultantů. Proměnná `retry` pak určuje, za kolik sekund po uplynutí `timeoutu` začne v případě vyzváněcí strategie `rrmemory` vyzvánět další linka v pořadí.

Proměnná `periodic-announce-frequency` určuje, po kolika sekundách se volajícímu přehraje oznámení uvedené v proměnné `periodic-announce`. Název oznámení je názvem zvukového souboru ve formátu `wav` (povinně 16 bitů, 8000 Hz, mono) bez přípony, uloženého v adresáři `/usr/local/share/asterisk/sounds`.

Pokud není do fronty přihlášen žádný konzultant, proměnná `joinempty` určuje, zda se má do ní hovor přesto zařadit, hodnota `leavewhenempty` pak určuje, zda má hovor takovou frontu opustit. V případě dynamické fronty dojde k opuštění fronty nejen při nepřítomnosti konzultantů, ale i při odhlášení posledního přihlášeného konzultanta.

Způsob řazení hovorů do jednotlivých front bude definován v nastavení vytáčecího plánu `extensions.conf`.

6.2.5 Vytvoření audio souborů s oznámeními

Ve vzorové konfiguraci jsou použita celkem 3 oznámení. Oznámení před vstupem do hlasové schránky při volání mimo pracovní dobu (`volno.wav`), oznámení před vstupem do fronty technické podpory (`podpora-1.wav`) a oznámení, které se přehrává každých 30 sekund během pobytu hovoru ve frontě (`podpora-2.wav`, viz též kapitolu 6.2.4 – proměnná `periodic-announce`).

Soubory jsou umístěny v adresáři `/usr/local/share/asterisk/sounds` a jsou součástí elektronické přílohy č. 2. Pro jejich vytvoření byl použit program Epos pro převod textu na hlas s nadstavbou Winsay (součást elektronické přílohy č. 6). V praxi je však vhodné nechat vytvořit jednotlivá oznámení ve studiu. Elektronický hlas by mohl některé zákazníky odradit od dalšího volání. Oznámení musí být nahráno ve formátu `wav`, 16 bitů, 8000 Hz, mono.

6.2.6 Nastavení vytáčecího plánu – extensions.conf

Vytáčecí plán je nejdůležitější součástí ústředny. Určuje, jakým způsobem budou zpracovávány požadavky na vytvoření komunikačního kanálu na základě volaného a v případě potřeby i volajícího čísla (extenze). Soubor `extensions.conf` je opět rozdělen do sekcí, které zde nazýváme kontexty (kromě sekcí `[general]` a `[globals]`). Části logiky vytáčecího plánu jsou pro některé kontexty společné, ty pak můžeme definovat v jiné sekci (nazýváme ji podkontextem) a tu zahrnout do základního kontextu pomocí direktivy `include =>`.

Začneme definicí sekce `[general]`:

```
[general]
writeprotect=yes
```

Sekce obsahuje pouze jedinou volbu a to `writeprotect=yes`. To znamená, že soubor `extensions.conf` nebude, při nechtěném použití příkazu CLI (Command Line Interface – příkazový řádek) ústředny `dialplan save`, přepsán (ústředna umožňuje úpravy vytáčecího plánu pomocí svého CLI, což však pro naše účely není podstatné, neboť konfigurace prostřednictvím textového souboru je přehlednější).

V sekci `[globals]` uvedeme pro zjednodušení budoucích úprav (např. změna operátora nebo změna telefonních čísel) proměnné, na které se budeme ve vytáčecím plánu odkazovat. Hodnotu proměnné potom získáme ve vytáčecím plánu použitím výrazu `${PROMENNA}`.

```
[globals]
OPERATOR=Operator
PROVOLBA=1231231
DELKA_PROVOLBY=7
DELKA_LINKY=2
TIMEOUT_FRONTY=1200
TIMEOUT_RECEPCE=20
TIMEOUT_OBCHOD=15
TIMEOUT_OSTATNI=30
```

Do proměnné `OPERATOR` uložíme jméno operátora, tak jak bylo definováno v souboru `sip_operator.conf`. V případě, že jméno v `sip_operator.conf` změníme, stačí pro účely vytáčekého plánu provést změnu pouze v této proměnné. Proměnná `PROVOLBA` obsahuje prefix provolby (v našem případě, kdy máme přidělenou sadu 100 čísel, je to prvních 7 číslic sady). `DELKA_PROVOLBY` je pak tedy 7 a `DELKA_LINKY` 2 (česká telefonní čísla obsahují 9 číslic).

Proměnná `TIMEOUT_FRONTY` určuje maximum doby, po kterou může být příchozí hovor ve frontě. `TIMEOUT_RECEPCE` určuje, jak dlouho bude vyzvánět telefon na recepci, `TIMEOUT_OBCHOD` určuje, jak dlouho budou vyzvánět telefony na telefonech jednotlivých obchodníků (jsou-li připojeni) a `TIMEOUT_OSTATNI` pak, jak dlouho budou vyzvánět telefony při přímém volání na jednotlivé linky (v případě, že tato čísla zveřejníme). Po uplynutí těchto časů proběhne další definovaná akce.

6.2.7 Syntaxe základních operátorů a funkcí

Nyní je nutno popsat syntaxi základních operátorů a funkcí použitých v našem vytáčekém plánu. Při zpracování hovorů pracujeme s tzv. extenzemi, což jsou v běžných telekomunikacích pouze číselné identifikátory linek. V případě Asterisku lze použít pro identifikátor i znaky anglické abecedy. My se však pro zjednodušení plánu budeme držet pouze číselných identifikátorů.

Syntaxe pro zpracování extenze pomocí operátoru `exten =>` je následující:

```
exten => cislo,priorita,aplikace()
```

Parametr `cislo` je číslo, pro které se má akce zadaná parametrem `aplikace()` provést. V hodnotě parametru lze použít zástupné znaky, přičemž to, že jsou použity, určuje znak `_` (podtržítka) na začátku hodnoty parametru. Základní zástupné znaky jsou tyto:

- X – jedno libovolné číslo od 0 do 9
- Z – jedno libovolné číslo od 1 do 9
- N – jedno libovolné číslo od 2 do 9

- [12-9] – jedno číslo z rozsahu uvedeného v hranatých závorkách (zde 1, 2 až 9)
- . (tečka) – jeden nebo více znaků bez ohledu na umístění
- ! (vykřičník) – žádný nebo více znaků bez ohledu na umístění

Například řetězec `_zx.` určuje, že číslo začíná na první pozici číslicí z rozsahu 1 až 9, na druhé pozici číslicí z rozsahu 0 až 9 a od třetí pozice může obsahovat libovolný počet libovolných číslic (v případě Asterisku i znaků abecedy). Řetězec `_[0-8]x` pak určuje, že číslo je dvouciferné s tím, že první číslice je z rozsahu 0 až 8, a druhá číslice z rozsahu 0 až 9. Parametr `priorita` určuje pořadí, ve kterém dojde k provedení akce určené parametrem `aplikace()`. Lze zadat buď číselnou hodnotu, nebo hodnotu `n`, která znamená `next` (další, neboli `priorita + 1`) Parametr `aplikace()` pak určuje akci, která nastane v případě, že právě zpracovávaná extenze odpovídá parametru `cislo`.

Požadujeme-li na dalším řádku zpracování další akce pro stejnou extenzi, použijeme operátor `same =>`, jehož syntaxe je následující:

```
same => priorita, aplikace()
```

Popis parametrů je shodný s popisem parametrů operátoru `exten =>`.

V některých níže uvedených funkcích se jako parametr `cislo` předává proměnná kanálu `EXTEN`, obsahující právě zpracovávanou extenzi. Její celou hodnotu získáme zadáním výrazu `${EXTEN}`. Požadujeme-li pouze její část, získáme ji zadáním výrazu `${EXTEN:x:y}`, kde `x+1` je pozice začátku části, kterou chceme získat a `y` je počet požadovaných číslic. Je-li parametr `y` vynechán, jsou vráceny všechny číslice od počátku `x+1` až do konce. Obsahuje-li proměnná `${EXTEN}` hodnotu například `123123150`, pak výraz `${EXTEN:7}` vrátí hodnotu `50` (je oříznuto prvních sedm číslic) a výraz `${EXTEN:3:5}` vrátí hodnotu `12315` (jsou oříznuty první tři číslice a vráceno pět následujících číslic).

```
GoTo(kontext, cislo, priorita)
```

Funkce `GoTo()` zajistí bezpodmínečný přechod na jiný kontext nebo podkontext na řádek s danou prioritou (pořadím) s tím, že v tomto kontextu bude zpracována extenze určená parametrem `cislo`. V našem případě budeme používat pro parametr `cislo` vestavěnou extenzi **s (start)**, která zjednodušeně řečeno zastupuje extenzi zpracovávanou před zavoláním funkce `GoTo()`.

```
GotoIfTime(casy,dny_v_tydnu,dny_v_mesici,mesice?navesti)
```

Funkce `GotoIfTime()` zajistí přechod na dané návěští v případě splnění časové podmínky určené těmito parametry (musí být splněny všechny podmínky zároveň):

`casy` – interval časů počatek–konec ve 24 hodinovém formátu. počatek a konec mohou nabývat hodnot 00:00 až 23:59. Například 08:00–18:00 znamená, že časová podmínka bude splněna v případě, že aktuální čas je mezi těmito hodnotami.

`dny_v_tydnu` – jeden či více dnů v týdnu, nebo interval `prvni_den-posledni_den`. Den může nabývat hodnoty `sun, mon, tue, wed, thu, fri` a `sat`. Hodnoty a rozsahy lze kombinovat pomocí operátoru `&`. Např. `wed&fri-sun` znamená, že podmínka bude splněna ve středu a v pátek až sobotu.

`dny_v_mesici` – jeden nebo více dnů v měsíci, či interval `prvni_den-posledni_den` a jejich kombinace. Den může nabývat hodnot 1 až 30. Např. `1-5&15-20` znamená, že podmínka bude splněna ve dnech 1-5 a 15-20 daného (či každého) měsíce.

`mesice` – jeden nebo více měsíců, případně interval `prvni_mesic-posledni_mesic` a jejich kombinace. Měsíc může nabývat hodnot `jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov` a `dec`. Např. `jan&jul-nov` znamená, že podmínka bude splněna v lednu a v červenci až listopadu.

Místo přesné hodnoty lze zadat znak `*`, který odpovídám všem možným hodnotám daného parametru.

```
Dial( typ/identifikátor ,casovy_limit ,volby)
```

Funkce `Dial()` s výše uvedenou zjednodušenou syntaxí iniciuje volání na zařízení s danou extenzí(identifikátorem). Parametry jsou následující:

`typ` – typ použitého kanálu – v našem případě SIP.

`identifikátor` – extenze zařízení, na které bude volání iniciováno.

`casovy_limit` – jak dlouho bude dané zařízení vyzvánět, než bude provedena další akce.

`volby` – voleb je nepřeborné množství, my použijeme pouze volby `T` a `t`. Volba `T` umožní volajícímu takzvaně slepě přesměrovat hovor na jinou linku pomocí klávesy `#`, volba `t` umožní totéž volanému. Slepé přesměrování znamená, že po stisknutí klávesy `#` a následném zadání cílové linky dojde k okamžitému přesměrování a hovor na původním zařízení je ukončen.

Pro další volby viz <http://www.voip-info.org/wiki/view/Asterisk+cmd+Dial>.

```
Playback( jmeno_souboru)
```

Funkce `Playback()` přehraje zvukový soubor zadaný parametrem `jmeno_souboru` bez přípony. Přehrávají se soubory z adresáře `/usr/local/share/asterisk/sounds`.

```
Set( promenna=hodnota)
```

Funkce `Set()` nastaví proměnnou `promenna` na hodnotu `hodnota`.

Např. `Set(PickUpNumber=${PROVOLBA}90)` nastaví proměnnou `PickUpNumber` na hodnotu `123123190` (proměnná `PROVOLBA=1231231`).

```
Queue( jmeno_fronty ,volby ,URL ,announceoverride ,casovy_limit)
```

Funkce `Queue()` přesměruje hovor do fronty s názvem `jmeno_fronty` definované v souboru `queues.conf` (viz kapitolu 6.2.4). Z voleb použijeme opět pouze hodnoty `T` a `t` již popsané výše u funkce `Dial()`. Pro další hodnoty parametru `volby` viz <http://www.voip-info.org/wiki/view/Asterisk+cmd+Queue>.

Parametr `URL` slouží k zadání URL, které bude posláno na volané zařízení, pokud toto umožňuje příjem URL. My tento parametr používat nebudeme.

Parametr `announceoverride` obsahuje jméno zvukového souboru s oznámením, které chceme přehrát volajícímu místo výchozího oznámení definovaného v souboru `queues.conf` v parametru fronty `periodic-announce`.

Parametr `casovy_limit` určuje maximální dobu v sekundách, po kterou bude hovor zařazen ve frontě. Po skončení této doby hovor frontu opustí a vytáček plán bude pokračovat další akcí definovanou za voláním funkce `Queue()`.

```
VoiceMail(cislo_schranky,volby)
```

Funkce `VoiceMail()` provede přeměrování hovoru do hlasové schránky. Parametr `cislo_schranky` určuje číslo schránky definované v souboru `voicemail.conf`. V parametru `volby` použijeme pouze hodnotu **s**, která zajistí, že se volajícímu po vstupu do schránky nebudou přehrávat žádné instrukce a on bude moci ihned nahrávat svůj vzkaz.

```
VoicemailMain(cislo_schranky)
```

Funkce `VoicemailMain()`, jejímž parametrem je číslo schránky, umožní volajícímu přihlásit se do hlasové schránky a vyslechnout si vzkazy v této schránce uložené.

```
PickUpChan(typ/identifikátor)
```

Funkce `PickUpChan()` umožní libovolnému zařízení převzít hovor přicházející na jiné zařízení (resp. vyzvánějící na jiném zařízení). Parametr `typ` určuje typ použitého kanálu (v našem případě `SIP`), `identifikátor` pak extenzi zařízení, jehož příchozí hovor chceme převzít. Např. výraz `PickUpChan(SIP/40)` zajistí převzetí hovoru přicházejícího na `SIP` telefon s linkou číslo 40.

```
AddQueueMember(jmeno_fronty)
```

```
RemoveQueueMember(jmeno_fronty)
```

Funkce `AddQueueMember()` zařadí volající extenzi do fronty podpory s názvem `jmeno_fronty`. Funkce `RemoveQueueMember()` ji naopak vyřadí.


```
ExecIf( ${podminka} ? funkce_pokud_ano() : funkce_pokud_ne() )
```

Funkce `ExecIf()` slouží k zavolání dané aplikace při splnění nebo nesplnění dané podmínky. Je-li podmínka splněna, je zavolána funkce `funkce_pokud_ano()`, jinak je zavolána funkce `funkce_pokud_ne()`. Například po zavolání

```
ExecIf( ${ "${AQMSTATUS}" = "ADDED" } ? Playback(agent-loginok) )
```

je v případě, že je extenze úspěšně přidána do fronty, přehráno přihlašujícímu se konzultantovi hlášení `agent-loginok`.

Funkce `Answer` bez parametrů – zajistí zvednutí linky a vytvoření kanálu před provedením dalších akcí.

Funkce `Hangup()` bez parametrů okamžitě ukončí zpracování vytáčeného plánu pro danou extenzi.

6.2.8 Popis logiky vzorového vytáčeného plánu

Kompletní vytáčený plán `extensions.conf`, jehož logiku si zde popíšeme, je součástí textové přílohy č. 6 a elektronické přílohy č. 2. Operátory, funkce a proměnné níže uvedené byly popsány v předchozí kapitole. Zde si tedy již popíšeme pouze chování plánu.

Začneme podkontextem `[vnitrni-sluzby]`, který slouží pro práci s hlasovou schránkou, frontou volajících a pro přebírání příchozích hovorů.

```
[vnitrni-sluzby]
exten => _**.,1,PickupChan(SIP/${EXTEN:${DELKA_LINKY}})

exten => 99,1,VoicemailMain(${CALLERID(number)})
    same => n, Hangup()

exten => 96,1,Answer
    same => n, AddQueueMember(DynamickaFrontaPodpory)
    same => n, ExecIf( ${ "${AQMSTATUS}" = "MEMBERALREADY" } ? Playback(agent-
alreadyon) )
    same => n, ExecIf( ${ "${AQMSTATUS}" = "ADDED" } ? Playback(agent-loginok) )
    same => n, Hangup()
```

```
exten=> 97,1,Answer
    same => n,RemoveQueueMember(DynamickaFrontaPodpory)
    same => n,Playback(agent-loggedoff)
    same => n,Hangup()
```

Obsluha první extenze `_**`. umožní v případě zadání znaků `**` na číselníku telefonu následovaných libovolnými číslicemi převzít hovor vyzvánějící na zařízení s extenzí, jež je dána číslicemi zadanými za `**`.

Obsluha druhé extenze `99` umožní volajícímu po zadání čísla `99` přihlášení ke hlasové schránce.

Obsluha třetí extenze `96` v prvním kroku zajistí vytvoření kanálu a v druhém kroku přidá volající extenzi do fronty podpory `DynamickaFrontaPodpory`. Pokud byl konzultant ve frontě již přihlášen, je mu to ve třetím kroku oznámeno, jinak je mu ve čtvrtém kroku oznámeno, že byl do fronty přidán. V pátém kroku je provádění vytáčecího plánu ukončeno.

Obsluha čtvrté extenze `97` opět v prvním kroku vytvoří kanál, ve druhém kroku konzultanta odhlásí z fronty podpory, ve třetím kroku mu odhlášení potvrdí a nakonec ukončí provádění plánu.

Podkontext `[prichozi-podpora]` má za úkol zpracovat hovory přicházející na číslo technické podpory `123 123 100` nebo `00`. Extenze `s` zde znamená, že se zpracovává číslo zpracovávané ve volajícím nadřazeném kontextu.

```
[prichozi-podpora]
exten => s,1,GotoIfTime(8:00-18:00,mon-fri,*,*?pracovni-doba)
    same => n,GotoIfTime(18:00-8:00,mon-fri,*,*?volno)
    same => n,GotoIfTime(*,sat-sun,*,*?volno)

    same => n(pracovni-doba),Answer
    same => n,Playback(podpora-1)
    same => n,Queue(DynamickaFrontaPodpory,tT,,,{TIMEOUT_FRONTY})
```

```

same => n,Queue(StatickaFrontaPodpory,tT,,,{TIMEOUT_FRONTY})
same => n,Hangup()

same => n(volno),Dial(SIP/40,{TIMEOUT_RECEPCE})
same => n,Answer
same => n,Playback(volno)
same => n,Voicemail(00,s)
same => n,Hangup()

```

První tři řádky (řádek z názvem podkontextu nepočítáme) určují na základě aktuálního času, jakým způsobem bude příchozí hovor zpracován. První řádek definuje, že v pracovní dny od 8:00 do 18:00 má další zpracování hovoru pokračovat od řádky s návěstím **pracovni-doba** (pátý řádek). Druhé dva řádky říkají, že v sobotu, v neděli a v době od 18:00 do 8:00 má zpracování pokračovat od řádku s návěstím **volno** (12. řádek).

V pracovní době (řádky 5-9) je vytvořen kanál, volajícímu je sděleno, že se dovolal na linku technické podpory a je zařazen do první fronty volajících (viz kapitolu 6.2.4). Pokud není v první dynamické frontě přihlášen žádný konzultant, nebo uplyne čas definovaný proměnnou `TIMEOUT_FRONTY` (v sekci `[globals]` je nastavena na 1200 sekund), je z fronty vyřazen a přesunut do druhé statické fronty, kde setrvá dalších `TIMEOUT_FRONTY` sekund. Po uplynutí této doby hovor statickou frontu opustí a provádění plánu i hovor jsou ukončeny.

V době volna (řádky 11-15) hovor nejprve vyzvání na lince 40 (například na recepci) po dobu určenou proměnnou `TIMEOUT_RECEPCE` (nastavena na 20 sekund), poté je vytvořen kanál a volajícímu přehráno oznámení, že se dovolal mimo pracovní dobu. Nakonec je přesměrován do hlasové schránky s číslem 00 a po zavěšení je zpracování plánu ukončeno.

Podkontext `[prichozi-obchod]` má za úkol zpracovat hovory přicházející na číslo obchodního oddělení 123 123 190.

```
[prichozi-obchod]
exten => s,1,GotoIfTime(8:00-18:00,mon-fri,*,*?pracovni-doba)
    same => n,GotoIfTime(18:00-8:00,mon-fri,*,*?volno)
    same => n,GotoIfTime(*,sat-sun,*,*?volno)

    same => n(pracovni-doba),Dial(SIP/40,${TIMEOUT_RECEPCE})
    same => n,Dial(SIP/41,${TIMEOUT_OBCHOD})
;... z důvodu úspory místa jsou čísla 42-58 vynechána
;... pro doplnění stačí zkopírovat předchozí řádek a upravit čísla
    same => n,Dial(SIP/59,${TIMEOUT_OBCHOD})
    same => n,Goto(prichozi-obchod,s,1)
    same => n,Hangup()

    same => n(volno),Dial(SIP/40,${TIMEOUT_RECEPCE})
    same => n,Answer
    same => n,Playback(volno)
    same => n,VoiceMail(90,s)
    same => n,Hangup()
```

První tři řádky určí stejným způsobem jako v podkontextu `[prichozi-podpora]` způsob zpracování hovoru v závislosti na čase. V pracovní době přejde zpracování na řádek s návěstím `pracovni-doba` (pátý řádek) a hovor vyzvání 20 sekund (proměnná `TIMEOUT_RECEPCE`) na lince č. 40. Poté postupně vyzvání 15 sekund (proměnná `TIMEOUT_OBCHOD`) na každé z linek 41-59, jsou li připojeny. Není-li některá z linek připojena, pokračuje vyzvánění bez jakékoliv prodlevy na další lince v pořadí. Po ukončení vyzvánění se zpracování hovoru vrací na začátek podkontextu a v cyklu pokračuje do doby, než je hovor přijat nebo ukončen volajícím.

V době volna (návěstí `volno`) je hovor zpracován stejným způsobem jako v podkontextu `[prichozi-podpora]` s tím, že číslo hlasové schránky je 90.

Podkontext [prichozi] určuje způsob zpracování hovorů přicházejících z ústředny VoIP operátora.

```
[prichozi]
exten => _${PROVOLBA}00,1,Goto(prichozi-podpora,s,1)
exten => _${PROVOLBA}90,1,Goto(prichozi-obchod,s,1)
exten  =>  _${PROVOLBA}XX,1,Dial(SIP/${EXTEN:${DELKA_PROVOLBY}},${
TIMEOUT_OSTATNI})
    same => n,Goto(prichozi-obchod,s,1)
```

Je-li příchozí volané číslo rovno hodnotě 123123100 (proměnná PROVOLBA je nastavena na hodnotu 1231231, linka je 00), je ve druhém řádku přeměřováno další zpracování hovoru na první řádek (s prioritou 1) již zmíněného podkontextu [prichozi-podpora]. Je-li číslo rovno hodnotě 123123190, je zpracování ve třetím řádku přeměřováno na první řádek podkontextu [prichozi-obchod]. Pokud číslo nekončí číslicemi 00 nebo 90, vyzvání hovor na konkrétním zařízení s extenzí rovnou dvěma posledním číslicím volaného čísla po dobu danou proměnnou TIMEOUT_OSTATNI (v sekci [globals] nastavena na hodnotu 30). Po uplynutí této doby je zpracování hovoru přeměřováno na první řádek podkontextu [prichozi-obchod].

Podkontext [vnitrni-pobocky] slouží ke zpracování hovorů mezi jednotlivými pobočkami.

```
[vnitrni-pobocky]
include => vnitrni-sluzby
exten => 00,1,Goto(prichozi-podpora,s,1)
exten => _[0-8]X,1,Dial(SIP/${EXTEN},,tT)
    same => n,Hangup()
```

První řádek říká, že součástí tohoto podkontextu má být i již definovaný podkontext [vnitrni-sluzby]. Druhý řádek má za úkol předat zpracování hovoru podkontextu [prichozi-podpora]. Toho využijeme především v případě, že se někdo nedopatřením dovolá na jinou linku a my jej chceme přepojit ať již naslepo nebo řízeně do fronty podpory. Třetí řádek zajistí, že při volání na čísla 01 až 89 bude hovor správně

nasměrován na telefon s danou vnitřní linkou (vnitropodnikové volání). Čtvrtý řádek zajistí ukončení zpracování plánu po zavěšení linky volaným nebo volajícím.

Podkontext **[blokovane-linky]** zajistí, aby uživatelé ústředny nemohli v případě odchozích hovorů volat na čísla placených služeb začínajících dvojčíslím 90.

```
[blokovane-linky]
exten => _90X.,1, Playback(pbx-invalid)
    same => n, Playback(vm-goodbye)
    same => n, Hangup()
```

V prvním a druhém kroku je při zadání čísla začínajícího dvojčíslím 90 přehrána volajícímu zpráva o neplatné extenzi (neplatném čísle). Ve třetím kroku je zpracování plánu ukončeno.

Vzhledem k tomu, že u odchozích hovorů poboček patřících do kontextů podpora, vyvoj a obchod je nežádoucí, aby se volaným účastníkům zobrazovalo číslo konkrétní pobočky, vytvoříme pro tyto kontexty dva podkontexty ([odchozi-callerid-100] a [odchozi-callerid-190]), které se starají o zpracování odchozích hovorů s tím, že modifikují číslo, které se bude na zařízení volaného zobrazovat. Tyto dva kontexty jsou shodné až na koncové dvojčíslí provolby, které má být v případě kontextu podpora rovno hodnotě 00 a v případě kontextu obchod hodnotě 90. Uvedeme si tedy pouze jeden z nich:

```
[odchozi-callerid-100]
exten => _00ZX.,1,Set(CALLERID(number)=${PROVOLBA}00)
    same => n,Dial(SIP/${EXTEN}@${OPERATOR})
    same => n, Hangup()

exten => _ZX.,1,Set(CALLERID(number)=${PROVOLBA}00)
    same => n,Dial(SIP/${EXTEN}@${OPERATOR})
    same => n, Hangup()
```

První a pátý řádek zajistí pomocí funkce `Set` nastavení vestavěné proměnné `CALLERID(number)` obsahující číslo zobrazované na přístroji volaného na hodnotu 123123100. Druhý a šestý řádek zajistí vytočení vnější linky prostřednictvím VoIP

operátora a třetí a sedmý řádek přeruší po ukončení hovoru zpracování vytáčecího plánu. Zda začne zpracování prvním nebo pátým řádkem závisí na volaném čísle. Pokud budeme volat číslo v mezinárodním formátu (např. 00420123123100), začne zpracování na prvním řádku. Pokud ne (číslo nebude začínat dvojčíslím 00), začne zpracování na řádku pátém.

U odchozích hovorů poboček patřících do kontextu ostatni se může volanému zobrazit číslo konkrétní pobočky. Pro tyto pobočky vytvoříme podkontext [odchozi-callerid-auto], jehož chování bude stejné jako u předchozího až na řádky 1 a 5, které budou vypadat takto:

```
exten => _00ZX.,1,Set(CALLERID(number)=${PROVOLBA}${CALLERID(number)})
...
exten => _ZX.,1,Set(CALLERID(number)=${PROVOLBA}${CALLERID(number)})
...
```

Je vidět, že výsledná hodnota proměnné CALLERID(number) je sestavena z hodnoty provolby (7 číslic) a současné hodnoty této proměnné (dvojciferné číslo pobočky). Výsledné zobrazené číslo bude tedy v případě pobočky 81 rovno 123123181.

Výše uvedené podkontexty popisovaly funkční část vytáčecího plánu. Na základě těchto podkontextů pak již jen sestavíme podkontexty pro definici odchozích hovorů, jejichž jediným úkolem je zahrnout v sobě stávající podkontexty, a výsledné kontexty jednotlivých linek. Podkontexty odchozích hovorů budou definovány následujícím způsobem:

```
[odchozi-obchod]
include => blokovane-linky
include => odchozi-callerid-190

[odchozi-vyvoj]
include => blokovane-linky
include => odchozi-callerid-190
```

```
[odchozi-podpora]
include => blokovane-linky
include => odchozi-callerid-100

[odchozi-ostatni]
include => blokovane-linky
include => odchozi-callerid-auto
```

Jak je vidět, jediné, co odlišuje tyto podkontexty, je nastavení rozdílných čísel zobrazovaných na přístroji volaného pomocí podkontextů `odchozi-callerid-*`.

Výsledné kontexty jsou nakonec definovány takto:

```
[obchod]
include => vnitрни-pobocky
include => odchozi-obchod
include => prichozι

[vyvoj]
include => vnitрни-pobocky
include => odchozi-vyvoj
include => prichozι

[podpora]
include => vnitрни-pobocky
include => odchozi-podpora
include => prichozι

[ostatni]
include => vnitрни-pobocky
include => odchozi-ostatni
include => prichozι
```

Opět je vidět jediný rozdíl, a to způsob zpracování odchozích hovorů pomocí podkontextů `odchozi-*`.

6.3 Nastavení telefonů, základní příkazy CLI ústředny

6.3.1 Nastavení telefonů

I když každý telefon, ať hardwarový či softwarový, má různá rozhraní pro nastavení připojení, konfigurace SIP má shodné základní parametry, i když rozdílně pojmenované. Těmi jsou: uživatelské jméno (Username, SIP User ID, Authorization User, Authenticate ID...), heslo (Password, Authenticate Password...), registrátor (Registrar, SIP Server, SIP Proxy...), UDP port serveru a zobrazované číslo (Display Name, Account Name...)

V případě naší ústředny by měly být tyto parametry nastaveny následujícím způsobem:

Uživatelské jméno – jméno zařízení v souboru `sip_*.conf`, například 60 v případě názvu [60].

Heslo - parametr `secret` v souboru `sip_sablony.conf` v šabloně [sablon-a-vse].

Registrátor – IP adresa telefonní ústředny

UDP port – 5060

Zobrazované číslo – je shodné se jménem zařízení v souboru `sip.conf` a tedy i uživatelským jménem.

6.3.2 Základní příkazy CLI ústředny

Pro účely aplikování změn a testování konfigurace, případně pro zobrazování stavu ústředny lze použít CLI (Command Line Interface – příkazový řádek) ústředny, do něhož se lze dostat zavoláním příkazu `asterisk -r` z příkazového řádku operačního systému a z něhož se lze dostat zavoláním příkazu `exit` z CLI ústředny. Na následující straně je uvedeno pouze několik základních příkazů, které využijeme.

Příkazy pro aplikování změn v konfiguračních souborech:

`reload` – načte novou konfiguraci ze všech změněných konfiguračních souborů.

`sip reload` – načte novou konfiguraci z konfiguračního souboru `sip.conf` a souborů v něm zahrnutých pomocí direktivy `#include`.

`queue reload all` – načte novou konfiguraci ze souborů `queues.conf` a `queuerules.conf`.

`voicemail reload` – načte novou konfiguraci ze souboru `voicemail.conf`.

`dialplan reload` – načte novou konfiguraci vytáčekého plánu ze souboru `extensions.conf`.

Příkazy pro sledování stavu ústředny:

`sip show peers` – zobrazí seznam připojených zařízení.

`sip show peer xx` – zobrazí informace o konkrétním zařízení s číslem `xx`.

`queue show` – zobrazí fronty volajících a konzultanty v nich přihlášené.

`core show channels` – zobrazí seznam aktivních kanálů/hovorů.

`core show channel ID_kanału` – zobrazí informace o konkrétním kanálu/hovoru se zadaným ID.

Příkazy pro řízení běhu ústředny:

`core stop now` – okamžitě ukončí činnost ústředny, v našem případě se však ústředna opět automaticky nainstaluje díky spuštěnému skriptu `astdog.sh` (viz kapitolu 7).

`core restart now` – okamžitě restartuje ústřednu a znovu načte všechna nastavení.

7. Závěr

Tématem práce byla problematika použití (nasazení) VoIP pomocí open source ústředny Asterisk.

Cílem práce bylo seznámit čtenáře jak s hlavními teoretickými základy souvisejícími s implementací VoIP, tak s možnostmi telefonní ústředny Asterisk a s požadavky na její bezproblémový provoz. Dalším cílem bylo umožnit IT administrátorům zavést v jejich podniku IP telefonii bez potřeby předchozích zkušeností s implementací VoIP a připravit pro ně základní vzorovou konfiguraci, ze které mohou vycházet a postupně ji rozvíjet, nebo ji rovnou použít pouze s drobnými úpravami nastavení.

Telefonní ústředna Asterisk je software s otevřeným kódem sponzorovaný firmou Digium. Vývoj Asterisku byl započat v roce 1999 panem Markem Spencerem a po letech vývoje se z ní stal i přes běžné chyby, které provázejí vývoj každého software, mocný nástroj poskytující mnohem více možností, než dosavadní telefonní ústředny závislé na běžném způsobu telefonie a na běžném telefonickém připojení. Asterisk mimo běžného přepojování hovorů umožňuje například ukládání příchozích hovorů do front, z nichž si je jednotliví operátoři vyzvedávají, práci s hlasovými schránkami, nahrávání hovorů, použití služby ENUM¹⁵ ¹⁶, dnes již málo používané faxování, interaktivní menu (lidově řečeno plechová huba) používané mnoha českými firmami (např. telefonní operátoři, banky, pojišťovny, poskytovatelé internetu) a další. Výhodou použití této ústředny (i každé jiné, která není licencována za úplatu) jsou nízké pořizovací náklady, neboť jediná investice spočívá v nákupu běžného počítače nebo serveru a mzdě zaměstnance provádějícího její konfiguraci. Další výhodou použití jakékoliv softwarové ústředny je, že v případě stěhování firmy v rámci telekomunikačním zákonem určeného území stačí přenést ústřednu a telefony do nového sídla, zapojit je do počítačové sítě, a bez jakékoliv zbytečné administrativy a časové prodlevy pokračovat se stejnými telefonními čísly. V případě, že je nutno změnit telefonní čísla (resp. provolbu telefonních čísel), stačí pro obnovení chodu ústředny malá časově nenáročná změna v její konfiguraci (cca 5 minut práce).

¹⁵ <http://www.nic.cz/enum/>

¹⁶ http://cs.wikipedia.org/wiki/Telephone_Number_Mapping

V práci uvedené teoretické základy i praktické návody nebyly zpracovány pouze na základě tištěné literatury [1] a [2], ale i na základě dalších informací vyhledaných pomocí Internetu (viz Použité zdroje) a na základě vlastních zkušeností získaných při tvorbě této práce, neboť ne vždy byly informace v uvedené literatuře úplné či přesné. To však nelze autorům uvedených zdrojů vyčítat, neboť i oni jsou mnohdy omezeni rozsahem své práce a časem, který mohou tématu věnovat.

V této práci bohužel nebylo možno díky jejímu plánovanému rozsahu popsat všechny možnosti telefonní ústředny Asterisk a tak se autor omezil pouze na popis základní funkcionality. I přesto byl plánovaný rozsah překročen. Budiž však omluvou, že dané téma je tak rozsáhlé, že nelze vynechat ani jednu z uvedených částí této práce. Zároveň je potřeba se omluvit za možné drobné nepřesnosti zde uvedené, neboť přes veškerou snahu získat správné informace nebylo možno v některých případech ověřit přesnost či pravdivost ve zdrojích uvedených informací. V těchto případech bylo nutno experimentovat a teprve na základě těchto experimentů vyvozovat závěry.

I přes drobné obtíže, které se vyskytly během tvorby této práce, lze prohlásit, že její cíl byl splněn, neboť čtenář bude schopen pochopit základní principy IP telefonie a navíc má k dispozici obecný návod na základní nastavení telefonní ústředny včetně kompletních konfiguračních souborů. Čtenáři-administrátorovi, který se rozhodne telefonní ústřednu nasadit a nebude chtít provádět žádné změny konfigurace, navíc stačí po drobných změnách konfigurace použít v elektronické příloze uloženou již nastavenou telefonní ústřednu.

8. Přílohy

8.1 Příloha č. 1 - základní instalace FreeBSD 9.1

Instalace by měla být minimalistická pouze se základními systémovými službami, zdrojovými kódy a dokumentací.

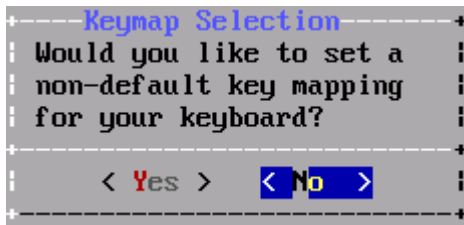
Konfigurace instalace probíhá prostřednictvím dialogů zobrazených instalátorem. V úvodním dialogu zvolíme položku *<Install>*.

Obrázek 3 – úvodní obrazovka instalátoru FreeBSD



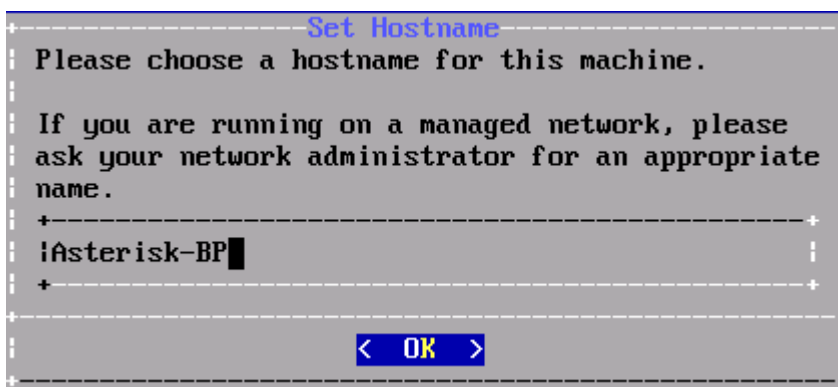
Pro účely této instalace není potřeba měnit rozvržení klávesnice.

Obrázek 4 – volba změny rozvržení klávesnice



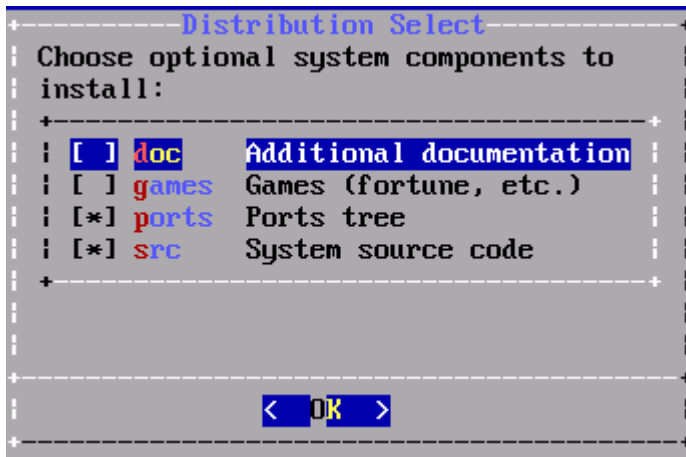
Počítač pojmenujeme tak, aby jeho název nekolidoval s názvem jiného, již existujícího.

Obrázek 5 – volba jména počítače



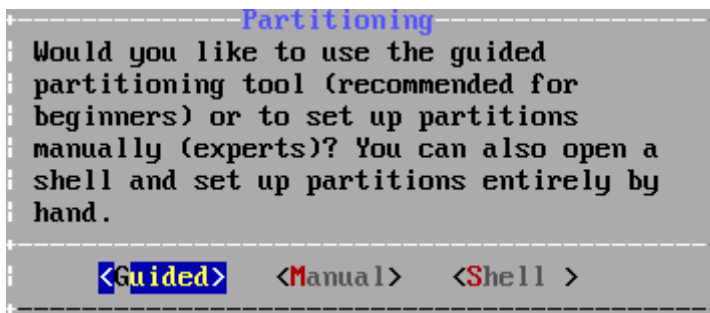
Z instalovaných součástí vybereme pouze *Ports tree* (kolekce portů) a *System source code* (zdrojové kódy systému pro případ nutnosti změny konfigurace jádra systému).

Obrázek 6 – volba instalovaných součástí



Pro vytvoření diskových oddílů použijeme průvodce.

Obrázek 7 – volba průvodce pro vytvoření oddílů



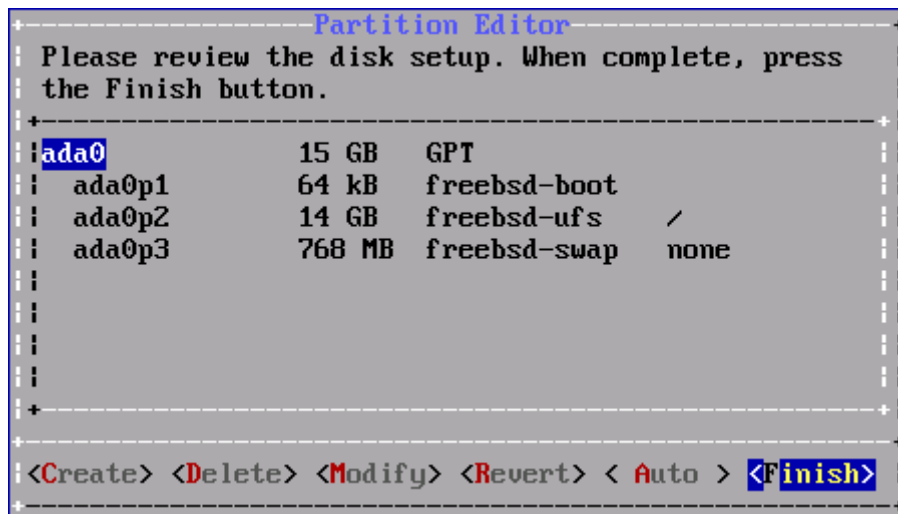
Pro systém použijeme celý disk.

Obrázek 8 – použití celého disku pro FreeBSD



Ponecháme systémem navržené výchozí rozdělení disku.

Obrázek 9 – výchozí nastavení rozdělení disku



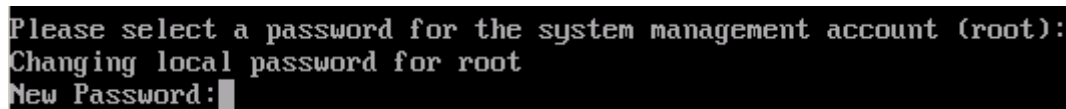
Nastavení potvrdíme.

Obrázek 10 – potvrzení nastavení rozdělení disku



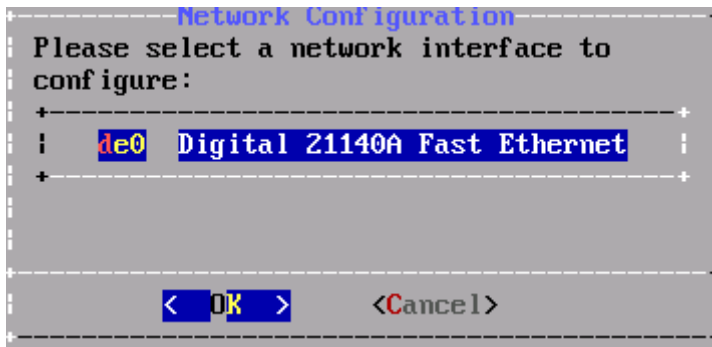
Po potvrzení předchozího dialogu dojde k vytvoření a naformátování oddílů a k rozbalení požadovaných balíčků na disk. Tato operace trvá přibližně 15 minut. Po dokončení této operace budeme požádáni o zadání hesla uživatele root.

Obrázek 11 – nastavení hesla uživatele root

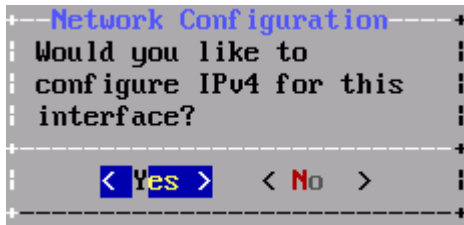


Virtuálnímu síťovému adaptéru, který je pod Hyper-V rozpoznán jako de0 a pod VirtualBoxem jako em0, nastavíme pevnou IP adresu (odmítneme DHCP konfiguraci).

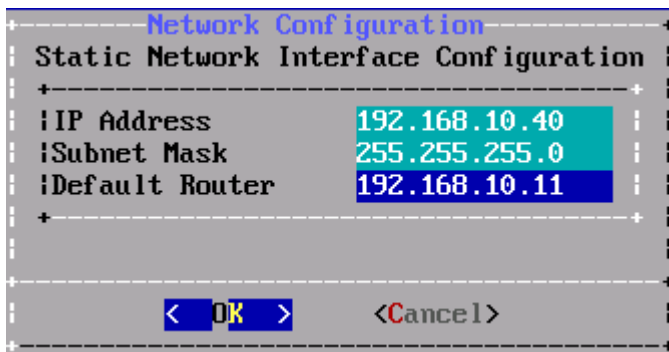
Obrázek 12 – volba síťového adaptéru



Obrázek 13 – volba nastavení IPv4 adresy

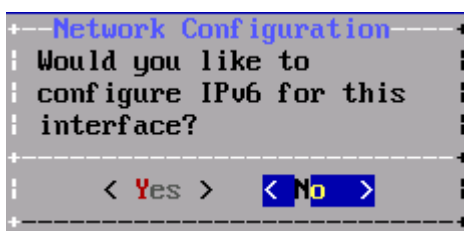


Obrázek 14 – nastavení pevné IP adresy



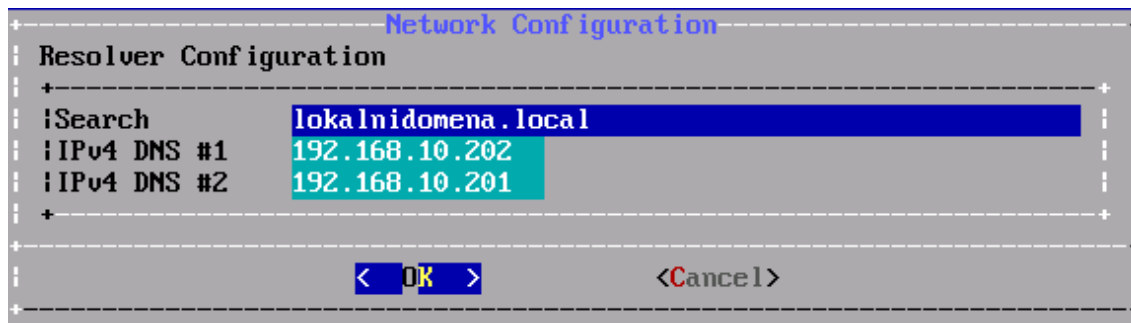
IPv6 adresu nebudeme nastavovat.

Obrázek 15 – volba nastavení IPv6 adresy



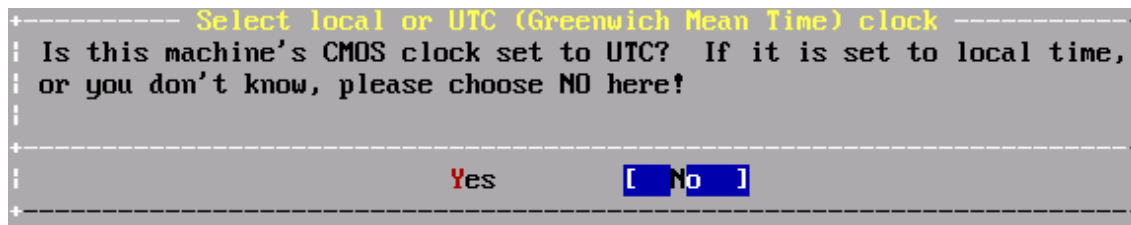
DNS nastavíme podle pravidel sítě, ve které bude systém provozován. Používáme-li v síti lokální doménu (např. lokalnidomena.local), zapíšeme její příponu do pole Search.

Obrázek 16 – nastavení DNS

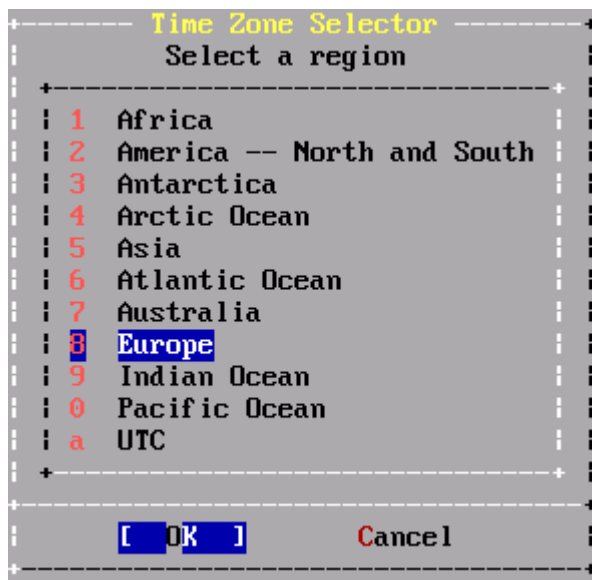


Časová zóna bude nastavena na CET (Central European Time – Středoevropský čas).

Obrázek 17 – nastavení časové zóny, 1. krok



Obrázek 18 – nastavení časové zóny, 2. krok



Obrázek 19 – nastavení časové zóny, 3. krok

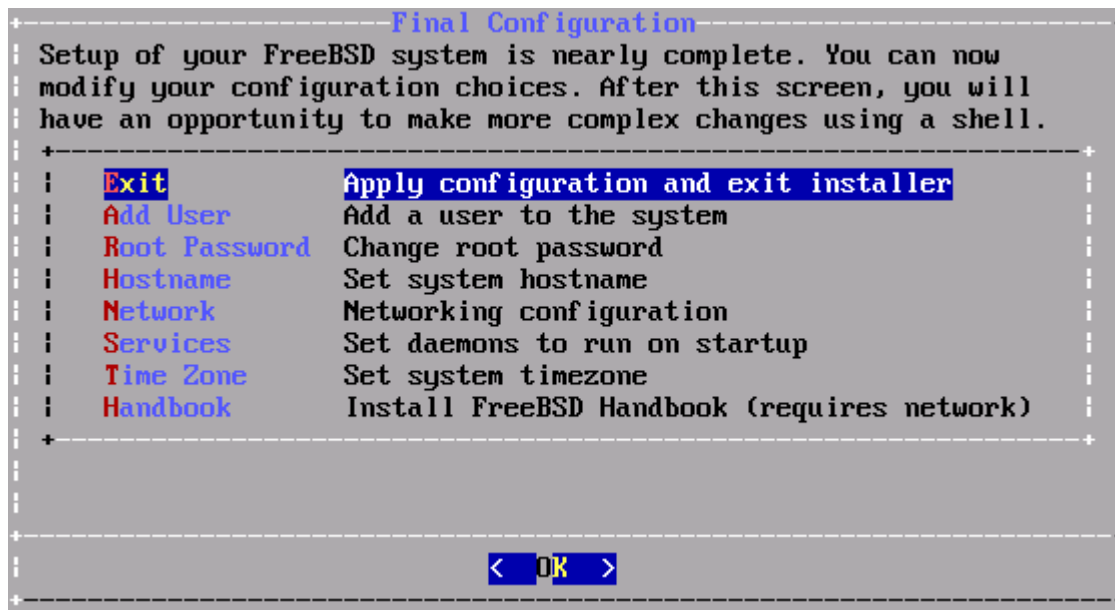


Obrázek 20 – nastavení časové zóny, 4. krok



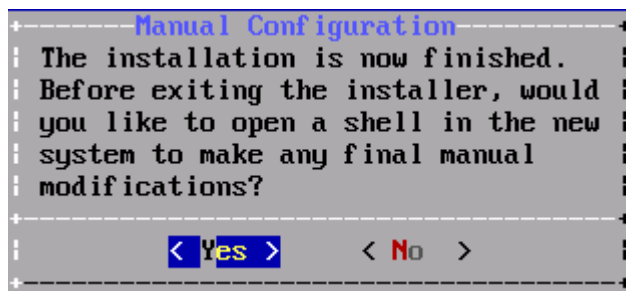
V následujícím dialogu dokončíme instalaci.

Obrázek 24 – dokončení instalace



Vzhledem k tomu, že před dalším startem právě nainstalovaného systému musíme odpojit instalační médium (nechceme, aby se při dalším spuštění zobrazil instalátor), zvolíme přechod do příkazového řádku.

Obrázek 25 – volba přechodu do příkazového řádku po dokončení instalace



Pokud instalujeme na fyzický počítač, vysuneme nejprve instalační médium a příkazem

```
init 6
```

systém restartujeme.

Pokud instalujeme na virtuální počítač, zadáme do příkazového řádku příkaz

```
init 0
```

který zajistí vypnutí počítače. Po odpojení instalačního média v konfiguraci virtuálního počítače můžeme systém opět spustit.

8.2 Příloha č. 2 - konfigurace systému a instalace dalších součástí

Chceme-li k systému přistupovat pomocí ssh (Secure Shell) jako uživatel `root`, je potřeba povolit jeho přihlášení v konfiguračním souboru `/etc/ssh/sshd_config` změnou hodnoty vlastnosti `PermitRootLogin` na `yes`. Po změně je nutno službu `sshd` restartovat pomocí příkazu:

```
/etc/rc.d/sshd restart
```

Jsme-li k systému připojeni vzdáleně prostřednictvím ssh (například pomocí programu Putty¹⁷), může dojít k nechtěné ztrátě spojení. V případě, že k takové situaci dojde a právě probíhá kompilace a instalace nějakého programu či jiná operace, je tato přerušena. Nechceme-li, aby k přerušení došlo, použijeme pro vzdálenou práci se systémem program `screen`¹⁸, který lze nainstalovat pomocí příkazu:

```
cd /usr/ports/sysutils/screen; make install clean
```

V zobrazeném konfiguračním dialogu ponecháme výchozí nastavení.

Preferujeme-li interpret příkazového řádku `bash`, je možno jej nainstalovat z kolekce portů pomocí příkazu:

```
cd /usr/ports/shells/bash; make install clean
```

V obou zobrazených konfiguračních dialozích ponecháme výchozí nastavení.

Po instalaci je nutno změnit výchozí interpret pomocí příkazu:

```
chsh -s /usr/local/bin/bash
```

¹⁷ <http://www.putty.org/>

¹⁸ <http://www.gnu.org/software/screen/>

Jsme-li zvyklí na správce souborů Midnight Commander (příkaz `mc`), nainstalujeme jej z kolekce portů pomocí příkazu:

```
cd /usr/ports/misc/mc; make install clean
```

Všechny konfigurační dialogy ponecháme v jejich výchozím nastavení.

Pro pravidelnou synchronizaci času každou hodinu například se serverem `ntp.cgi.cz` je nutno přidat do souboru `/etc/crontab` řádek:

```
0 * * * * root /usr/sbin/ntpdate ntp.cgi.cz
```

8.3 Příloha č. 3 – konfigurační soubor `modules.conf`

```
[modules]
autoload=yes
;Vynucené moduly
load => res_musiconhold.so
;Nutno zavest explicitne, jinak nebude fungovat fronta
load => app_queue.so
;
;Námi zakázané moduly
noload => pbx_gtkconsole.so
noload => chan_alsa.so
noload => chan_console.so
noload => chan_dahdi.so
noload => chan_gtalk.so
noload => chan_h323.so
noload => chan_jingle.so
noload => chan_mgcp.so
noload => chan_misdn.so
noload => chan_multicast_rtp.so
noload => chan_nbs.so
noload => chan_oo323.so
noload => chan_phone.so
noload => chan_skinny.so
noload => chan_unistim.so
noload => chan_usbradio.so
noload => chan_vpb.so
noload => app_adsiprog.so
noload => app_alarmreceiver.so
noload => app_amd.so
noload => app_chanisavail.so
noload => app_dahdibarge.so
```

```
noload => app_dahdiras.so
noload => app_db.so
noload => app_getcpeid.so
noload => app_ivrdemo.so
noload => app_jack.so
noload => app_milliwatt.so
noload => app_morsecode.so
noload => app_readfile.so
noload => app_rpt.so
noload => app_skel.so
noload => app_waitforring.so
noload => codec_adpcm.so
noload => codec_dahdi.so
noload => codec_g722.so
noload => codec_g726.so
noload => codec_ilbc.so
noload => codec_lpc10.so
noload => codec_speex.so
noload => format_g723.so
noload => format_g726.so
noload => format_g729.so
noload => format_h263.so
noload => format_h264.so
noload => format_ilbc.so
noload => format_jpeg.so
noload => format_siren14.so
noload => format_siren7.so
noload => format_sln16.so
noload => format_vox.so
noload => cdr_tds.so
noload => cdr_adaptive_odbc.so
noload => cdr_odbc.so
noload => cdr_manager.so
noload => cdr_custom.so
noload => cdr_syslog.so
noload => cdr_csv.so
```

8.4 Příloha č. 4 – konfigurační soubor /etc/rc.conf

```
hostname="asterisk-bp"
#konfigurace sítě pro Hyper-V
#ifconfig_de0=" inet 192.168.10.40 netmask 255.255.255.0"
#defaultrouter="192.168.10.1"
#konfigurace sítě pro VirtualBox
ifconfig_em0=" inet 192.168.10.40 netmask 255.255.255.0"
defaultrouter="192.168.10.1"
sshd_enable="YES"
moused_enable="YES"
asterisk_enable="YES"
asterisk_flags="-p"
dumpdev="NO"
```

8.5 Příloha č. 5 – kompletní soubor sip.conf

```
#include sip_sablony.conf

[general]
context=bez-registrace ; Vychodi kontext
allowguest=yes ; Povoleni hovoru z neregistrovanых zarizeni
udpbindaddr=0.0.0.0 ; Ustredna bude naslouchat na libovolnem
                    ; sitovem rozhrani
disallow=all ; Zakazeme vsechny kodeky
allow=alaw ; Povolime pouze kodek alaw
allowoverlap=no ; Disable overlap dialing support

#include sip_operator.conf
#include sip_obchod.conf
#include sip_vyvoj.conf
#include sip_podpora.conf
#include sip_ostatni.conf
```

8.6 Příloha č. 6 - kompletní soubor extensions.conf

```
[general]
writeprotect=yes

[globals]
OPERATOR=Operator ;jmeno operatora uvedene v sip_operator.conf
PROVOLBA=1231231
DELKA_PROVOLBY=7 ;pocet cislic provolby
DELKA_LINKY=2 ;pocet cislic vnitřni linky
TIMEOUT_FRONTY=1200 ;velikost timeoutu fronty podpory v sec.
TIMEOUT_RECEPCE=20 ;velikost timeoutu recepce v sec.
TIMEOUT_OBCHOD=15 ;velikost timeoutu jednotlivých obchodníků v sec.
TIMEOUT_OSTATNI=30 ;velikost timeoutu ostatních poboček v sec.

;-----
;kontexty vnitřních linek
;-----

[bez-registrace]
exten => _X.,1,Goto(prichozi-podpora,s,1)

[obchod]
include => vnitřni-pobocky
include => odchozi-obchod
include => prichozi

[vyvoj]
include => vnitřni-pobocky
include => odchozi-vyvoj
include => prichozi

[podpora]
include => vnitřni-pobocky
include => odchozi-podpora
```



```

include => prichozi

[ostatni]
include => vnitri-pobocky
include => odchozi-ostatni
include => prichozi

;-----
;volani vnitri-nich pobocek na vnitri pobocky
;-----
[vnitri-pobocky]
include => vnitri-sluzby

;prepojzeni do fronty podpory
exten => 00,1,Goto(prichozi-podpora,s,1)
;volani na ostatni vnitri linky
;(vnitri linky 9X slouzi pro vnitri sluzby)
exten => _[0-8]X,1,Dial(SIP/${EXTEN},,tT)
    same => n,Hangup()

;-----
;odchozi hovory
;-----
[odchozi-obchod]
include => blokovane-linky
include => odchozi-callerid-190

[odchozi-vyvoj]
include => blokovane-linky
include => odchozi-callerid-190

[odchozi-podpora]
include => blokovane-linky
include => odchozi-callerid-100

[odchozi-ostatni]
include => blokovane-linky
include => odchozi-callerid-auto

[blokovane-linky]
exten => _90X.,1, Playback(pbx-invalid)
    same => n, Playback(vm-goodbye)
    same => n, Hangup()

[odchozi-callerid-190]
exten => _00ZX.,1, Set(CALLERID(number)=${PROVOLBA}90)
    same => n, Dial(SIP/${EXTEN}@${OPERATOR})
    same => n, Hangup()

exten => _ZX.,1, Set(CALLERID(number)=${PROVOLBA}90)
    same => n, Dial(SIP/${EXTEN}@${OPERATOR})
    same => n, Hangup()

[odchozi-callerid-100]
exten => _00ZX.,1, Set(CALLERID(number)=${PROVOLBA}00)
    same => n, Dial(SIP/${EXTEN}@${OPERATOR})
    same => n, Hangup()

exten => _ZX.,1, Set(CALLERID(number)=${PROVOLBA}00)

```

```

same => n,Dial(SIP/${EXTEN}@${OPERATOR})
same => n,Hangup()

[odchozi-callerid-auto]
exten => _00ZX.,1,Set(CALLERID(number)=${PROVOLBA}${CALLERID(number)})
same => n,Dial(SIP/${EXTEN}@${OPERATOR})
same => n,Hangup()

exten => _ZX.,1,Set(CALLERID(number)=${PROVOLBA}${CALLERID(number)})
same => n,Dial(SIP/${EXTEN}@${OPERATOR})
same => n,Hangup()

;-----
;prichozi hovory
;-----
[prichozi]
exten => _${PROVOLBA}00,1,Goto(prichozi-podpora,s,1)
exten => _${PROVOLBA}90,1,Goto(prichozi-obchod,s,1)

exten => _${PROVOLBA}XX,1,Dial(SIP/${EXTEN}:${DELKA_PROVOLBY}),${
TIMEOUT_OSTATNI})
same => n,Goto(prichozi-obchod,s,1)

[prichozi-podpora]
exten => s,1,GotoIfTime(8:00-18:00,mon-fri,*,*?pracovni-doba)
same => n,GotoIfTime(18:00-8:00,mon-fri,*,*?volno)
same => n,GotoIfTime(*,sat-sun,*,*?volno)

same => n(pracovni-doba),Answer
same => n,Playback(podpora-1)
same => n,Queue(DynamickaFrontaPodpory,tT,,${TIMEOUT_FRONTY})
same => n,Queue(StatickaFrontaPodpory,tT,,${TIMEOUT_FRONTY})
same => n,Hangup()

same => n(volno),Dial(SIP/40,${TIMEOUT_RECEPCE})
same => n,Answer
same => n,Playback(volno)
same => n,Voicemail(00,s)
same => n,Hangup()

[prichozi-obchod]
exten => s,1,GotoIfTime(8:00-18:00,mon-fri,*,*?pracovni-doba)
same => n,GotoIfTime(18:00-8:00,mon-fri,*,*?volno)
same => n,GotoIfTime(*,sat-sun,*,*?volno)

same => n(pracovni-doba),Dial(SIP/40,${TIMEOUT_RECEPCE})
same => n,Dial(SIP/41,${TIMEOUT_OBCHOD})
same => n,Dial(SIP/42,${TIMEOUT_OBCHOD})
same => n,Dial(SIP/43,${TIMEOUT_OBCHOD})
same => n,Dial(SIP/44,${TIMEOUT_OBCHOD})
same => n,Dial(SIP/45,${TIMEOUT_OBCHOD})
same => n,Dial(SIP/46,${TIMEOUT_OBCHOD})
same => n,Dial(SIP/47,${TIMEOUT_OBCHOD})
same => n,Dial(SIP/48,${TIMEOUT_OBCHOD})
same => n,Dial(SIP/49,${TIMEOUT_OBCHOD})
same => n,Dial(SIP/50,${TIMEOUT_OBCHOD})
same => n,Dial(SIP/51,${TIMEOUT_OBCHOD})
same => n,Dial(SIP/52,${TIMEOUT_OBCHOD})
same => n,Dial(SIP/53,${TIMEOUT_OBCHOD})

```

```

same => n,Dial(SIP/54,${TIMEOUT_OBCHOD})
same => n,Dial(SIP/55,${TIMEOUT_OBCHOD})
same => n,Dial(SIP/56,${TIMEOUT_OBCHOD})
same => n,Dial(SIP/57,${TIMEOUT_OBCHOD})
same => n,Dial(SIP/58,${TIMEOUT_OBCHOD})
same => n,Dial(SIP/59,${TIMEOUT_OBCHOD})
same => n,Goto(prichozi-obchod,s,1)
same => n,Hangup()

same => n(volno),Dial(SIP/40,${TIMEOUT_RECEPCE})
same => n,Answer
same => n,Playback(volno)
same => n,VoiceMail(90,s)
same => n,Hangup()

;-----
;vnitri sluzby
;-----
[vnitri-sluzby]
;Zvednuti hovoru vyzvanejiciho na jinem telefonu.
;PickupChan funguje jen pro konkretni typ kanalu, je potreba jej
definovat pro vsechny pouzivane typy.
exten => _*.*,1,PickupChan(SIP/${EXTEN:${DELKA_LINKY}})

;hlasova schranka
exten => 99,1,VoicemailMain(${CALLERID(number)})
    same => n,Hangup()

;prihlaseni operatora podpory do fronty
exten=> 96,1,Answer
    same => n,AddQueueMember(DynamickaFrontaPodpory)
    same => n,ExecIf("${AQMSTATUS}"="MEMBERALREADY"?Playback(agent-
alreadyon))
    same => n,ExecIf("${AQMSTATUS}"="ADDED"?Playback(agent-loginok))
    same => n,Hangup()
;odhlaseni operatora podpory z fronty
exten=> 97,1,Answer
    same => n,RemoveQueueMember(DynamickaFrontaPodpory)
    same => n,Playback(agent-loggedoff)
    same => n,Hangup()

```

9. Obsah elektronické přílohy

- [1] Bakalářská práce ve formátu PDF včetně tezí (adresář bp)

- [2] Adresář s konfiguračními soubory uvedenými v této práci ve stromové struktuře souborového systému FreeBSD (adresář asterisk)

- [3] ISO obraz instalačního média FreeBSD komprimovaný metodou zip (soubor FreeBSD-9.1-RELEASE-i386-disc1.zip)

- [4] VHD disk se vzorovou instalací komprimovaný metodou zip (soubor asterisk.zip).
VHD disk lze připojit pod jakýmkoliv virtualizačním prostředím, funkčnost je ověřena pouze pod Hyper-V a VirtualBoxem. Heslo uživatele root je 123456. Před použitím je nutno upravit názvové servery v souboru `/etc/resolv.conf` a nastavení sítě v souboru `/etc/rc.conf`. Dále je nutno upravit konfigurační soubor ústředny `extensions.conf`, v němž je potřeba změnit hodnotu proměnné `PROVOLBA` v sekci `[globals]` na hodnotu přidělenou VoIP operátorem, a konfigurační soubor `sip_operator.conf`, v němž je nutno změnit přihlašovací údaje k VoIP operátorovi.

- [5] Softwarové telefony Phoner - <http://www.phoner.de/>
a X-Lite - <http://www.counterpath.com/x-lite.html> (adresář sw_phones)

- [6] Software pro převod textu na hlas Epos s nadstavbou Winsay -
<http://epos.ure.cas.cz/> (adresář oznameni)

Použité zdroje:

- [1] Madsen, L., Meggelen, J.V.: Asterisk: the definitive guide. 3rd ed. Sebastopol, CA: O'Reilly Media, Inc, 2011. ISBN 978-059-6517-342
- [2] Kabelová, A., Dostálek, L.: Velký průvodce protokoly TCP/IP a systémem DNS, 2008, ISBN 978-80-251-2236-5
- [3] Peterka, Jiří.: Z historie internetové telefonie (nejen) v ČR [online].
URL: <<http://www.earchiv.cz/b06/b0401006.php3>>
- [4] Digium®, Inc.: Internetové stránky projektu Asterisk [online].
URL: <<http://www.asterisk.org/>>
- [5] Bělka, Jiří.: Začínáme bezpečně s FreeBSD [online]. [cit. 2004-08-03].
URL: <http://www.root.cz/clanky/zaciname-bezpecne-s-freebsd-7>
- [6] Wikipedie.: Network address translation [online].
URL: http://en.wikipedia.org/wiki/Network_address_translation
- [7] Křenek, Michal.: Slovník ABC Linuxu [online].
URL: <http://www.abclinuxu.cz/slovník/nat>
- [8] Wikipedie.: Real-time Transport Protocol [online].
URL: http://en.wikipedia.org/wiki/Real-time_Transport_Protocol
- [9] Wikipedie.: TCP [online].
URL: http://cs.wikipedia.org/wiki/Transmission_Control_Protocol
- [10] Wikipedie.: UDP [online].
URL: <http://cs.wikipedia.org/wiki/UDP>
- [11] Wikipedie.: SIP [online].
URL: http://cs.wikipedia.org/wiki/Session_Initiation_Protocol
- [12] Wikipedie.: SIP [online].
URL: http://en.wikipedia.org/wiki/Session_Initiation_Protocol
- [13] Voip-Info.org.: IAX [online].
URL: <http://www.voip-info.org/wiki/view/IAX>
- [14] Wikipedie.: G.711 [online].
URL: <http://en.wikipedia.org/wiki/G.711>
<http://cs.wikipedia.org/wiki/G.711>

- [15] Asterisk Guru.: [online].
URL: <http://www.asteriskguru.com>
- [16] VoIP Info.: [online].
URL: <http://www.voip-info.org>
- [17] Hruška, Petr.: Konfigurace Asterisku (7) – User, peer a friend [online].
URL: <http://www.telegro.cz/2009/05/23/konfigurace-asterisku-7-user-peer-friend>

Seznam tabulek a obrázků

Tabulka 1 – průběh hovoru mezi dvěma účastníky	14
Tabulka 2 – vodítko pro určení požadavků na hardware	20
Obrázek 1 - nastavení síťového adaptéru pod Hyper-V	25
Obrázek 2 – Nastavení síťového adaptéru pod VirtualBoxem.....	26
Obrázek 3 – úvodní obrazovka instalátoru FreeBSD.....	61
Obrázek 4 – volba změny rozvržení klávesnice	61
Obrázek 5 – volba jména počítače	61
Obrázek 6 – volba instalovaných součástí.....	62
Obrázek 7 – volba průvodce pro vytvoření oddílů	62
Obrázek 8 – použití celého disku pro FreeBSD.....	62
Obrázek 9 – výchozí nastavení rozdělení disku	63
Obrázek 10 – potvrzení nastavení rozdělení disku	63
Obrázek 11 – nastavení hesla uživatele root	63
Obrázek 12 – volba síťového adaptéru	64
Obrázek 13 – volba nastavení IPV4 adresy.....	64
Obrázek 14 – nastavení pevné IP adresy.....	64
Obrázek 15 – volba nastavení IPV6 adresy.....	64
Obrázek 16 – nastavení DNS	65
Obrázek 17 – nastavení časové zóny, 1. krok	65
Obrázek 18 – nastavení časové zóny, 2. krok	65
Obrázek 19 – nastavení časové zóny, 3. krok	66
Obrázek 20 – nastavení časové zóny, 4. krok	66
Obrázek 21 – volba služeb automaticky spouštěných po startu systému.....	67
Obrázek 22 – zakázání ladících výpisů.....	67
Obrázek 23 – odmítnutí vytvoření uživatelských účtů	67
Obrázek 24 – dokončení instalace	68
Obrázek 25 – volba přechodu do příkazového řádku po dokončení instalace	68