

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA
PROVOZNĚ EKONOMICKÁ FAKULTA

Katedra informačního inženýrství



Bakalářská práce
Administrace OS Solaris

Vypracovala: Veronika Janovská

Vedoucí bakalářské práce: doc. Ing. Arnošt Veselý, CSc.

© 2009

Schválené zadání s razítkem děkana

Čestné prohlášení

Prohlašuji, že bakalářskou práci na téma „Administrace OS Solaris“ jsem vypracovala samostatně pouze s použitím literatury a pramenů uvedených v seznamu citované literatury.

V Praze dne dubna 2009

Veronika Janovská

Poděkování

Děkuji vedoucímu bakalářské práce doc. Ing. Arnoštovi Veselému, CSc. za pozornost věnovanou této práci a za cenné rady, připomínky a metodické vedení práce.

**Bakalářská práce na téma
Administrace OS Solaris**

**Bachelor Thesis on Subject
Solaris OS Administration**

Anotace

Tato bakalářská práce pojednává o problematice administrace Solarisu, síťového operačního systému UNIXového typu. Cílem práce je nastítnit postupy používané při základní konfiguraci jeho nejnovější verze Solaris 10 a pomocí stručného přehledu o možnostech jeho využití poukázat na tento operační systém jako na vhodnou alternativu pro podniková řešení. Práce je složena ze tří logických celků.

V první, teoretické části, jsou zpracovány dosavadní poznatky z oblasti využití UNIXových operačních systémů. Jsou zde uvedeny základní charakteristiky systému a nastíněn historický kontext.

Praktická část obsahuje popis a ukázkou použití některých příkazů používaných k administraci operačního systému a implementaci některých služeb. Je zde uveden i návrh skriptu napsaný v příkazovém interpretu Bash, který by měl sloužit jako nástroj pro automatické zakládání uživatelských účtů.

V závěrečné části jsou zmíněny možné oblasti nasazení serverů s operačním systémem Solaris, uvedeny a zhodnoceny příklady z praxe.

Klíčová slova

UNIX, kernel, shell, distribuce, metadata, balíček, proces, služba, SMF,

Synopsis

This bachelor thesis handles Solaris, the UNIX-based network operating system, management issue. The objective is to mention the procedures used for the basic Solaris 10, the latest operating system version, configuration and to highlight this operation system as a suitable alternate for enterprise solutions by the means of its possible utilization concise survey. The thesis comprises three logical units.

There is compiled knowledge of the unix operating system utilization area up to now in the first notional part. Fundamental system features are introduced and historic context is outlined in this part.

The following practical part is comprised of the swatches of some administrative tasks and some services implementation. There is also a Bash script design presented which should serve as a tool for an automatic user account setup.

There are mentioned possible areas of deployment of servers with Solaris 10 OS installed in the third part. Some production examples are introduced and evaluated.

Keywords

UNIX, kernel, shell, distribution, metadata, package, process, service, SMF

Obsah

1	ÚVOD	6
2	CÍL PRÁCE A METODIKA	6
3	LITERÁRNÍ REŠERŠE	7
3.1	PRINCIPY OPERAČNÍHO SYSTÉMU UNIX	8
3.1.1	<i>Jádro operačního systému</i>	8
3.1.2	<i>Příkazový interpret</i>	8
3.1.3	<i>Hierarchický systém souborů</i>	9
3.1.4	<i>Procesy</i>	10
3.2	HISTORIE OPERAČNÍHO SYSTÉMU UNIX	11
3.2.1	<i>Multics a počátky UNIXu</i>	11
3.2.2	<i>Nástroj pro úpravu textů</i>	12
3.2.3	<i>Kernel v C</i>	12
3.2.4	<i>Komerční UNIX</i>	13
3.2.5	<i>Univerzitní UNIX</i>	13
3.3	OPERAČNÍ SYSTÉM FIRMY SUN MICROSYSTEMS.....	14
3.3.1	<i>Společnost a její portfolio</i>	14
3.3.2	<i>SunOS a Solaris</i>	15
3.4	SYSTEM V vs. BSD.....	16
4	SPECIFIKACE HARDWARE POUŽITÉHO PŘI ZPRACOVÁNÍ SYSTÉMU	17
4.1	HOSTNAME: MACEK	17
4.2	HOSTNAME: SUNRAY	17
4.3	RODINA PROCESORŮ ULTRASPARC II.....	18
5	ADMINISTRACE OPERAČNÍHO SYSTÉMU	18
5.1	INSTALACE, BALÍČKY A ZÁPLATY	19
5.1.1	<i>Instalace operačního systému</i>	19
5.1.2	<i>Práce s balíčky</i>	20
5.1.3	<i>Záplaty</i>	21
5.2	SPUŠTĚNÍ A ZASTAVENÍ OPERAČNÍHO SYSTÉMU	22
5.2.1	<i>Zapnutí a vypnutí operačního systému</i>	22
5.2.2	<i>Boot proces</i>	24
5.2.3	<i>Service Management Facility</i>	24
5.3	SPRÁVA DISKŮ, SOUBOROVÉ SYSTÉMY A SPRÁVA DATOVÝCH SVAZKŮ	26

5.3.1	Nástroj <i>format</i>	26
5.3.2	Práce se souborovými systémy	28
5.3.3	Správa datových svazků	29
5.3.4	Zálohování a obnova dat	30
5.4	SPRÁVA UŽIVATELŮ	31
5.4.1	Uživatelé a vytváření uživatelských účtů	31
5.4.2	Bezpečnost	33
5.4.3	Návrh skriptu pro hromadné vytváření uživatelů	34
5.5	SÍŤE	36
5.5.1	Konfigurace síťových rozhraní	36
5.5.2	Jmenné a adresářové služby	37
6	VÝKON A JEHO MONITOROVÁNÍ	39
6.1	SLEDOVÁNÍ VÝKONU	39
6.1.1	<i>Iostat, vmstat, netstat</i>	39
6.1.2	<i>Dtrace</i>	40
6.2	SPRÁVA ZDROJŮ	40
7	OBLASTI VYUŽITÍ OS SOLARIS	42
8	ZÁVĚR	44
9	SEZNAMY	46
9.1	LITERATURA A OSTATNÍ ZDROJE	46
9.2	OBRÁZKY	48
9.3	TABULKY	48
10	PŘÍLOHY	50
	PŘÍLOHA Č. 1 - SKRIPT PŘÍKAZOVÉHO INTERPRETU BASH PRO DÁVKOVÉ VYTVOŘENÍ UŽIVATELSKÝCH ÚČTŮ ZE SEZNAMU UŽIVATELŮ	50

1 Úvod

Ve snaze udržet si pozici na trhu jsou podniky nuceny hledat nové, efektivnější formy řízení a využívání zdrojů. Mezi oblasti s vysokým potenciálem optimalizace zdrojů patří i podniková informatika, resp. její technologická infrastruktura, která je tvořena technickými prostředky a jejich základním programovým vybavením. Právě do skupiny základního programového vybavení se řadí i operační systém.^[4]

Operační systém tvoří rozhraní mezi aplikacemi a hardware, se kterým komunikuje pomocí ovladačů. Dále zajišťuje správu procesů, tj. jejich vytváření, zpracování a vzájemnou komunikaci.^[12] Operační systém zabezpečuje i přidělování systémových zdrojů procesům, jako např. přístup ke vstupně výstupním zařízením, procesorový čas nebo vnitřní paměť.

Tato práce pojednává o Solaris 10, nejnovější verzi operačního prostředí vystavěné na operačním systému SunOS 5.10 založeném na bázi UNIX. Protože autorka se zabývá administrací tohoto operačního prostředí, je zde kladen důraz na jeho provozní aspekt.

2 Cíl práce a metodika

Účelem této práce je přiblížení operačního systému Solaris, aktuálně verze 10, jako vhodné alternativy pro serverová řešení podniků. Práce by měla posloužit k praktickému osvojení v ní popsaných, často používaných příkazů a k porozumění uvedeným technologiím. Text zmiňuje způsob správy operačního systému v grafickém prostředí, ale zaměřuje se především na použití příkazové řádky.

Přestože podpora pro platformu x86, konkrétně pro procesory Intel, byla již součástí operačního systému Solaris 2.1, k demonstraci v praktické části této práce je použit dvouprocesorový server Sun Enterprise 220R a nízkonákladová, plně 64-bitová pracovní stanice Sun Blade 150 firmy Sun Microsystems. Tento hardware založený na platformě SPARC, resp. UltraSPARC byl určen pro nepřetržitý provoz. Systémy SPARC používají obecně procesory s nižší frekvencí, ale naproti tomu vysokorychlostní sběrnice.^[13] Právě pro tyto počítače byl operační systém Solaris navržen a vykazuje na

nich nejlepší výkon. Práce se tedy zaměří na specifika systémové administrace na platformě SPARC.

Text se soustředí na oblasti správy, které se od ostatních UNIXových systémů liší. Úvodem do problematiky je kapitola o instalaci, jejích metodách a balíčkovacím systému. Dále je popsán start a zastavení operačního systému, kde důležitou roli plní Open Boot PROM firmware. Je zde zmíněn Service Management Facility, nový model správy systémových služeb, který byl představen v Solaris 10. V práci je zařazena i kapitola o správě datových úložišť a archivaci.

Součástí kapitoly věnované správě uživatelů je odladěný a otestovaný návrh skriptu napsaný v příkazovém interpretu Bash, který slouží pro automatické vytváření většího počtu uživatelských účtů. Příkazový interpret je velmi pružným nástrojem a jeho prostřednictvím lze řešit úkoly, které by např. v grafickém prostředí nebyly proveditelné. Následuje stručné pojení o sítích, jehož součástí je i kapitola o jmenných a adresářových službách. Činnosti, která je důležitá obzvláště pro konsolidovaná prostředí s více aplikacemi nasazenými na jednom fyzickém serveru, je věnována kapitola o sledování výkonu a správě zdrojů. Závěrem jsou zmíněny příklady nasazení operačního systému Solaris 10 v praxi.

Při zpracování tématu byly použity referenční manuály, materiály z webových prezentací firmy Sun Microsystems a manuálové stránky, které jsou součástí operačního systému.

3 Literární řešerše

Tato kapitola se soustředí na základní principy UNIXových operačních systémů a uvádí historické souvislosti jeho vývoje. Dále je zde nastíněn vývoj společnosti Sun Microsystems a jejího operačního systému a v závěru kapitoly následují charakteristiky dvou hlavních „proudů“ UNIXu System V a BSD.

3.1 Principy operačního systému UNIX

3.1.1 Jádro operačního systému

Základem každého operačního systému je jádro neboli kernel. Operační systémy typu UNIX mají jádro napsané v programovacím jazyku C.

Jedná se o program, který spravuje systémové zdroje. Jádro tvoří rozhraní mezi aplikacemi a technickými prostředky počítače a poskytuje klíčové systémové služby jako je správa procesů, souborů, virtuální paměti, vstupně výstupních zařízení apod.

Jádro operačního systému Solaris je modulární, tj. složené z objektů, které mohou podle potřeby být dynamicky nahrány do paměti. Kód jádra je rozdělen na dvě části, a to *genunix* nezávislý na platformě a *unix*, který je pro každou platformu specifický.^[13] Při bootu systému jsou tyto dvě části a další moduly nahrány do operační paměti, kde vytvoří fungující jádro systému.

Prostřednictvím systémových služeb umožňuje jádro komunikaci programového vybavení s technickými prostředky počítače. Služby systému jsou realizovány softwarovým přerušením^[12], kdy proces vykoná funkci volání jádra a vstoupí do privilegovaného režimu. Následuje identifikace požadované služby a je vykonána část kódu, kde je služba umístěna.

S technickými prostředky jádro komunikuje pomocí tzv. ovladačů, které mohou být do jádra v operační paměti načteny dynamicky, tj. podle potřeby.

3.1.2 Příkazový interpret

Příkazový interpret neboli *shell* slouží ke vzájemné komunikaci uživatele s operačním systémem. Bývá spuštěn procesem *getty* při úspěšném přihlášení uživatele do systému. Přijímá a zpracovává příkazy, které uživatel zadá do jeho textového rozraní, tzv. příkazové řádky. Pro příkazovou řádku má příkazový interpret definovanou sadu proměnných prostředí, které jsou na rozdíl od ostatních proměnných uvedeny ve verzálcích. K zadání příkazu je pro uživatele v příkazové řádce zobrazena tzv. výzva. Její forma může být definována proměnnou prostředí *PS1* a *PS2*.

Shell je rovněž skriptovacím jazykem, který umožňuje neinteraktivní zpracování příkazů. Kromě vnitřních a vnějších příkazů využívá účinné mechanismy, jako je

přesměrování, roura, expanzní znaky, proměnné a výrazy. Ukázka použití shellu jako skriptovacího jazyka je uvedena v praktické části této práce.

Jako standardní příkazový interpret pro UNIX se vžil Bourne shell, ale je možné zvolit z více druhů jako je např. Bourne-again shell (Bash) nebo C shell.

3.1.3 Hierarchický systém souborů

Souborové systémy v operačních systémech typu UNIX umožňují uspořádání souborů do logické struktury. Soubor je nestrukturovanou posloupností bytů a v systému souborů je reprezentován i-uzlem. Tato datová struktura je nositelem metadat souboru. I-uzel tedy poskytuje informace o:

- typu souboru
- přístupových právech
- individuálním vlastníku souboru
- skupině vlastníka souboru
- čase poslední modifikace souboru
- čase posledního přístupu do souboru
- čase poslední modifikace metadat souboru
- velikosti souboru v blocích
- počtu odkazů na soubor
- adresách datových bloků souboru.^[12]

V UNIXových souborových systémech je možné pracovat se třemi typy souborů. Obyčejné soubory obsahují uživatelská data. Pokud jsou to textové soubory, obsahují znaky ze spodní části tabulky ASCII a pro jejich výpis je možné použít příkazy *cat* nebo *more*. V binárních souborech se mohou vyskytovat všechny znaky z tabulky ASCII. Obsah těchto souborů je možné vypsát pomocí příkazu *od* a jeho přepínačů.

Dalším typem souborů jsou adresáře. Jsou to binární soubory, které obsahují seznam i-uzlů a názvů souborů. Obsah adresáře lze zjistit pomocí příkazu *ls* a jeho přepínačů.

Posledním typem souborů jsou speciální soubory. Tyto soubory zajišťují přístup uživatelů a procesů k technickým zdrojům poskytovaným výpočetním systémem. K perifériím a ostatním technickým prostředkům je zajištěn přístup dvojího druhu, a to

sice znakový neboli sekvenční a blokový. Typickým znakovým zařízením je terminál a bloková zařízení reprezentuje např. pevný disk. Speciální soubory jsou organizovány v adresáři */dev*. Soubory pro sekvenční přístup k zařízením v úplném výpisu příkazu *ls -l* (jedná se o parametr „*l*“) mají atribut *c* a soubory blokových zařízení atribut *b*.

Souborový systém vytvořený na diskovém oddílu je rozdělen do několika bloků, z nichž každý plní určité funkce. *Superblok* obsahuje metadata souborového systému, tj. údaje o velikosti souborového systému v blocích, jeho jménu a stavu, velikosti logického bloku, čase poslední úpravy, velikosti oblasti *i-uzlů* a jejich seznam. Bez superbloku je souborový systém nečitelný. *I-uzly* se vytvářejí při zakládání souborového systému a v *oblasti i-uzlů* je pro ně vyhrazen prostor s pevnou velikostí.^[12] Na počtu *i-uzlů* určeném velikostí systému souborů a velikostí vyhrazené položky závisí počet souborů v souborovém systému. Ostatní prostor souborového systému je vyhrazen pro datovou oblast, ve které jsou uloženy datové bloky souborů a adresářů.

Hierarchické uspořádání systému souborů je zajištěno pomocí odkazu na nadřazený adresář identifikovatelný jako „*...*“ a aktuální adresář označený jako „*..*“ v každém adresáři souborového systému.

3.1.4 Procesy

Zatímco program je spustitelný soubor, proces je jeho běžící instancí.^[11] Preemptivní multitasking v operačních systémech typu UNIX zajišťuje současný běh více procesů. Je tedy možné spustit více instancí jednoho programu. Pro identifikaci v systému běžících procesů slouží jednoznačný identifikátor PID, pomocí kterého je možno s procesem komunikovat. Volání jádra umožňuje procesům vytvářet nové procesy, ukončovat je, synchronizovat běh procesů a ošetřit příchod signálů.

Také procesy mají svoji hierarchii.^[11] Při zavedení operačního systému do paměti je spuštěn proces *swapper* s PID 0, který kromě zajištění správy paměti spustí další proces. V tomto případě je to proces *init* s identifikačním číslem 1, který vytváří pomocí volání služby *fork()* nové procesy potomky. V operační paměti je při zavedení operačního systému umístěna i tabulka procesů obsahující informace o spuštěných procesech. Zůstává v paměti po celou dobu běhu operačního systému. Informace jako stav procesu nebo priorita jsou pro operační systém klíčové.^[12] Naproti tomu informace

o pracovním adresáři, návratovém kódu procesu nebo o terminálu a otevřených souborech je možné v datové struktuře uživatelská oblast odložit, pokud proces neběží, na disk. Tyto dvě datové struktury, ve kterých si systém ukládá metadata procesů jsou součástí systémového kontextu procesu. Kromě toho existuje pro každý proces ještě uživatelský kontext, ve kterém je obsah jeho adresového prostoru – textu programu, dat a zásobníku.^[12]

Každý proces má svého vlastníka a jeho skupinu. Po přihlášení uživatele do systému získá proces login jeho identifikační číslo uživatele UID a číslo skupiny uživatele GID. Potomci, kteří jsou spuštěni procesem login, nesou stejné UID a GID. Proces tak může pracovat se soubory, k nimž má nastavena přístupová práva. Tento mechanismus může být obejit pomocí řízeného předávání uživatelských práv, kdy proces získá pomocí nastavených s-bitů vlastnické a skupinové identifikační číslo souboru. Takto je např. umožněna změna hesla neprivilegovaným uživatelům.

3.2 Historie operačního systému UNIX

V raném věku výpočetní techniky bylo možné s počítači komunikovat pouze prostřednictvím binárního kódu. S dynamickým rozvojem tohoto odvětví byl takový způsob komunikace neúnosný, a tak programátoři začali s vývojem programovacích jazyků. Dvojková soustava byla při komunikaci s počítačem nahrazena příkazy ve formě osmičkových či šestnáctkových čísel a později i slov. Takový programovací jazyk se skládal ze souboru příkazů a překladače, který zajistil interpretaci příkazů do strojového kódu. S postupem času bylo nutné vytvořit program, který by zajistil základní funkce systému a usnadnil tak programátorům práci. Na počátku 60. let 20. století začaly vznikat operační systémy.^[7] Nicméně každý počítač měl svůj vlastní systém s omezeným množstvím úloh, který obvykle sloužil k jedinému účelu.

3.2.1 Multics a počátky UNIXu

Jako jednu z příčin počátku vzniku operačního systému UNIX lze označit účast americké telekomunikační společnosti AT&T, resp. jejího výzkumného střediska Bell Laboratories na neúspěšném projektu Multics. Cílem projektu bylo vyvinout operační systém, který by umožňoval v reálném čase sdílet výkon a datová úložiště tehdejších

sálových počítačů více uživatelů. Ačkoliv se podařilo Multics nasadit na sálový počítač GE 645 společnosti General Electric, zdaleka nespĺňoval původní požadavky.^[1] Bell Laboratories od projektu ustoupily a ten byl v roce 1969 ukončen.

Snaha o vylepšení vývojového prostředí programů vedla pány Kena Thompsona a Dennise Ritchieho, programátory z Bell Laboratories, kteří se zúčastnili i projektu Multics, k vypracování návrhu předchůdce UNIXového souborového systému. Jeho chování nasimulovali v prostředí počítače GE 645, který byl společností Bell Laboratories pro projekt Multics původně zkoupen. Pro Multics Ken Thompson napsal počítačovou hru Space Travel. Kvůli špatnému ovládní a vysokým nárokům na výkon byla hra portována na počítač PDP-7 výrobce DEC. Tento minipočítač měl dobrý grafický displej a v té době ležel v Bell Laboratories nevyužitý.

Vývojové prostředí PDP-7 ovšem neumožňovalo konečné sestavení Space Travel, který byl napsán v assembleru, a tak bylo nutné na GE 645 vyrobit pásky a teprve pak je přenést na PDP-7.^[9] Tato skutečnost přiměla Thompsona a Ritchieho implementovat na počítač PDP-7 jejich dřívější návrh. Kromě jejich verze souborového systému nové vývojové prostředí obsahovalo implementaci subsystému procesů a několik systémových nástrojů. Nový systém byl v narážce na Multics nazván UNIX.

3.2.2 Nástroj pro úpravu textů

Operační systém UNIX byl na PDP-7 pouhou emulací v operační paměti, protože minipočítač neměl pevný disk. Navíc byl jednouživatelský a jeho jediným programovým vybavením byl loader a assembler.

Až požadavek na systém pro zpracování textu v patentovém oddělení Bell Laboratories znamenal zakoupení nového hardware. V roce 1970 byl tedy UNIX portován na počítač PDP-11/20, s pevným diskem o kapacitě 512 KB a 12 KB operační paměti vyhrazenými pro systém a 8 KB pro uživatelské programy.^[13]

3.2.3 Kernel v C

UNIX byl napsán v assembleru, který byl špatně čitelný a manipulovatelný. Proto se Thompson pokoušel o implementaci překladače pro Fortran, ale namísto toho vyvinul jazyk B. Tento jazyk byl velmi pomalý, proto Ritchie na přelomu 60. a 70. let

vytvořil jazyk C. V roce 1972 bylo jádro UNIXu přepsáno do tohoto jazyka. Obsahovalo cca 1000 řádků kódu v C a cca 1000 řádků kódu v assembleru a plnilo opravdu nejzákladnější funkce.

3.2.4 Komerční UNIX

K růstu popularity operačního systému přispěl článek publikovaný Ritchiem a Thompsonem v roce 1974 a v roce 1978 byla uveřejněna UNIX verze 7. Operační systém měl 40 KB, hierarchický souborový systém, plnou verzi překladače pro programovací jazyk C. a příkazový interpret Bourne shell. Tato verze byla portována na minipočítače Interdata 8/32 a sálové počítače IBM 360. Společnost AT&T začala vydávat její komerční licence a snažila se omezit výuku strojového kódu.

Integrací několika verzí operačního systému vyvíjeného v Bell Laboratories vznikl UNIX System III, ze kterého přidáním sady systémových nástrojů vznikl UNIX System V. Podporu System V Release 1 ohlásila společnost AT&T v roce 1983. Dalším vývojem bylo dosaženo Release 3, na kterém je vystavěn operační systém HP-UX resp. AIX společnosti Hewlett-Packard resp. IBM.

3.2.5 Univerzitní UNIX

Vzhledem k dohodě podepsané v roce 1965 mezi vládou Spojených států amerických a společností AT&T, ve které byla zakotvena omezení podnikatelských aktivit telekomunikačního monopolu, byly licence na zdrojový kód operačního systému UNIX udělovány pouze vzdělávacím institucím. Ze stejných důvodů nemohla být inzerována vydání aktualizací operačního systému a ani poskytována externí systémová podpora.^[2]

Protože zdrojový kód operačního systému byl na univerzitách dostupný, vznikala na jejich půdě řada projektů. Významnou se v tomto ohledu stala University of California v Berkeley. Tam byla ohlášena v roce 1974 distribuce známá jako BSD, která byla spíš rozšířením UNIXu verze 6 než samostatným operačním systémem. Tato distribuce obsahovala mimo jiné i překladač programovacího jazyka Pascal a ranou verzi textového editoru ex.^[14] Vývoj BSD distribuce probíhal na počítači PDP-11.

Komeracionalizace UNIXu společnosti AT&T přiměla vývojáře BSD distribuce vydat vlastní plnou verzi operačního systému. Za tímto účelem byla BSD distribuce portována na počítač s architekturou VAX. Protože port systému neumožňoval plné využití vlastností nové architektury, byla tedy do téměř přepsaného jádra přidána lepší implementace virtuální paměti. V roce 1979 byl pod označením 3BSD vydán kompletní operační systém, podle nové vlastnosti znám také jako VMUNIX. Úspěch této verze znamenal pro vývojáře BSD distribuce možnost zúčastnit se projektu rozvoje internetu, který byl sponzorován společností DARPA.

Na počátku 90. let minulého století byl projekt BSD na univerzitě v Berkeley ukončen. Dosažené výsledky byly uvolněny pod BSD licenci, na níž staví řada následovníků jako je např. FreeBSD. Stalo se tak v důsledku soudních sporů o proprietární zdrojový kód společnosti AT&T.

3.3 Operační systém firmy Sun Microsystems

3.3.1 Společnost a její portfolio

Vznik společnosti Sun Microsystems je spojován se čtyřmi doktorandy ze Stanfordské univerzity v Paolo Altu a je datován do roku 1982. Příslušnost zakladatelů k této vzdělávací instituci se odráží ve slově Sun, které je zkratkou projektu Stanford University Network.

Vize „Síť je počítačem“TM, která nese myšlenku, že celek je „šikovnější“ než jeho jednotlivé komponenty, umožnila společnosti stát se poskytovatelem výkonného technického a programového vybavení a služeb pro provoz internetu. Technologie Sun Microsystems se v posledních desetiletích staly jeho nedílnou součástí. Dnes už není výjimkou setkat se s nimi v mobilních telefonech, komplexních datových skladech a superpočítačích.^[6]

První UNIXová pracovní stanice byla navržena spoluzakladatelem společnosti Andy Bechtolsheimem, který usiloval o využití akademických objevů v komerční sféře. Pro technické prostředky firmy Sun Microsystems byla vyvinuta procesorová platforma SPARC založená na architektuře RISC. Rok po založení, tj. v roce 1983, získává společnost kontrakt na dodávku UNIXových serverů pro společnost Computervision.

Z této zakázky je hrazen další výzkum, a tak si Sun Microsystems v roce 1987 zajišťují prvenství v prodeji pracovních stanic. O dva roky později byl představen model SPARCstation 1, jehož komponenty byly naskládány do skříně velikostí a tvarem připomínající krabici na pizzu. Za další tři roky byl představen první multiprocessorový počítač tohoto typu SPARCstation10.^[6]

V 90. letech minulého století Sun Microsystems zaznamenal prvenství ve výkonnosti pracovních stanic a rovněž jeden milion prodaných systémů. V roce 1994 byl do kin uveden film Toy Story, který je kompletně vyhotoven prostředky počítačové animace právě na systémech Sun.

Jako technologií s největším možným dopadem na IT strategie podniku byl v roce 2006 oceněn systém SUN Fire T2000 určený pro provoz webových aplikací a distribuovaných databází. Systém s procesorem UltraSPARC T1 s frekvencí 9,6 GHz také označovaným NIAGARA je založen na architektuře SPARC verze 9, takže i na tomto serveru může být díky binární kompatibilitě spuštěn operační systém Solaris^[18].

Do portfolia společnosti Sun Microsystems patří kromě operačního systému Solaris i další softwarové produkty. Za zmínku stojí především souborový systém Network File System, který se díky bezplatné licenci stal standardem pro sdílení souborů v síti. Dalším významným produktem je objektově orientovaný programovací jazyk Java, jenž našel uplatnění v aplikacích vyvíjených pro internet, např. internetových obchodech, programech pro komunikaci v reálném čase nebo aplikacích pro mobilní telefony.

3.3.2 SunOS a Solaris

Operační systém SunOS vznikl jako derivát interní verze distribuce 4.1 BSD. S jeho vývojem započal Bill Joy – rovněž absolvent Stanford University. První verze operačního systému byla vydána v roce 1983 a na pracovních stanicích Sun brzy našel využití v prostředích pro počítačovou grafiku.^[2]

V 80. letech minulého století se rovněž Bell Laboratories v součinnosti se Sun Microsystems pokoušeli o sloučení obou větví UNIXu, tj. System V a BSD. V důsledku těchto aktivit je od verze SunOS 5.x operační systém firmy Sun Microsystems založen na UNIXové distribuci System V.^[13] Další vývoj operačního systému znamená uvolnění

pro platformu x86 a se Solaris 10 i pro 64-bitovou architekturu této platformy. V rámci Open Solaris existuje i port na architekturu PowerPC.

Pro pořádek ve jmenných konvencích je nutno uvést, že název SunOS je používán pro samotné jádro operačního systému a systém označovaný jako Solaris obsahuje i grafické uživatelské rozhraní a další zejména síťové aplikace.

Solaris byl vydán až s verzí SunOS 4.x, která byla založena na BSD distribuci UNIXu. SunOS 5.x a Solaris 2 mají již základy distribuce System V Release 4.^[28]

Je také možno říct, že označení SunOS je používáno pro operační systémy před přechodem na System V a název Solaris je používán pro mladší verze.

Zatímco v Solaris 2.6 je grafické rozhraní Common Desktop Environment, součástí Solaris 10 je kromě CDE i Java Desktop Environment grafické rozhraní založené na GNOME.

3.4 System V vs. BSD

Verze 7 operačního systému UNIX se stala společnou základnou všem současným operačním systémům UNIXového typu. Vývoj operačního systému UNIX se od tohoto okamžiku vydal dvěma hlavními směry, kdy proprietární System V byl volbou pro velké víceuživatelské systémy a BSD distribuce dominovala v oblasti pracovních stanic.^[27]

Když na počátku 90. let minulého století probíhaly soudní pře o zdrojový kód UNIXu, byly z distribuce BSD odstraněny všechny závislosti na operačním systému z Bell Laboratories. Výsledkem dalšího vývoje byl např. vývoj systému virtuální paměti, rychlý souborový systém s podporou dlouhých jmen a symbolických odkazů, základních prvků rodiny síťových protokolů TCP/IP s podporou Telnet a FTP. Mezi produkty vývojářské skupiny BSD distribuce se řadí kontrola úkolů, což je rozšíření systému správy procesů, standardní knihovny pro správu terminálů curses a termcap, nebo C-shell, který je dnes standardním příkazovým interpretem v distribucích založených na BSD. Naproti tomu BSD distribuce adoptovala např. technologii NFS, která vznikla v Sun Microsystems.

BSD skupina byla taktéž první u zrodu komunit, které se vysokou měrou podílejí na vývoji volně šiřitelného software.

Z distribuce System V je v operačním systému společnosti Sun Microsystems integrováno mnoho znaků. Jde např. o podporu meziprocesové komunikace, nové funkce shellu, knihovny pro práci v síti STREAMS a TLI, vzdálené sdílení souborů a propracované stránkování paměti.^[13]

4 Specifikace hardware použitého při zpracování systému

4.1 Hostname: macek

K ukázkám administrátorských úkonů v praktické části byl zvolen server Sun Enterprise 220R. Tento server je osazen dvěma procesory typu UltraSPARC-II s frekvencí 450 MHz, které zaručují jeho vysokou výkonnost. Velikost jeho operační paměti činí 2 GB, což znamená plně osazené všechny sloty paměťovými moduly o velikosti 128 MB. Systém je osazen dvěma UltraSCSI pevnými disky o kapacitě 36 GB, tj. celkem 2 x 36 GB. Pevné disky lze díky technologii hot-plug měnit za provozu systému. Jako standardní rozhraní má server na základní desce integrovanou síťovou kartu typu Ethernet/Fast Ethernet s přenosovou rychlostí 10 nebo 100 Mb/sec, dva sériové porty a jeden paralelní port. Na rozšiřující sběrnici nechybí čtyři PCI sloty. Server je určen pro instalaci do serverového regálu.^[23]

Tabulka 1: Zkrácený výpis systémové konfigurace serveru macek

```
root@macek / # prtdiag -v |head
System Configuration: Sun Microsystems sun4u Sun Enterprise 220R (2
X UltraSPARC-II 450MHz)
System clock frequency: 113 MHz
Memory size: 2048 Megabytes
...
(výpis přerušen)
```

4.2 Hostname: sunray

Přestože je Solaris spíše serverovým operačním systémem, k demonstraci uvedených administrátorských úkonů byl použit i systém Sun Blade 150 Workstation firmy Sun Microsystems. Jedná se o pracovní stanici s plně 64-bitovou architekturou, která je osazena procesorem UltraSPARC III s frekvencí 650MHz, operační paměť o velikosti 512 MB s podporou až 2GB a jedním pevným IDE diskem o kapacitě 74 GB

s podporou až 160 GB. Mezi standardní rozhraní patří na základní desce integrovaná síťová karta typu Ethernet/Fast Ethernet s přenosovou rychlostí 10 nebo 100 Mb/sec, jeden devítipinový sériový port a jeden paralelní pětadvacetipinový port. Na rozšiřující sběrnici se nachází tři 32-bitové PCI sloty.^[15]

Tabulka 2: Zkrácený výpis systémové konfigurace pracovní stanice sunray

```
root@sunray # prtdiag -v
System Configuration: Sun Microsystems sun4u Sun Blade 150
(UltraSPARC-IIe 650MHz)
System clock frequency: 93 MHZ
Memory size: 512MB
...
(výpis přerušen)
```

4.3 Rodina procesorů UltraSPARC II

UltraSPARC-II je druhá generace vysoce výkonných a integrovaných superskalárních procesorů vyvinutých firmou Sun Microsystems implementujících 64 bitovou RISC architekturu založenou na otevřeném standardu SPARC verze 9. Základní instrukční sada je rozšířena o instrukce zrychlení multimediálních informací VIS.^[5]

Procesor UltraSPARCII byl určen pro nízkorozpočtové pracovní stanice. Na čipu procesoru je integrovaná L2 cache o velikosti 512 KB, která dovoluje snížení latence operační paměti.

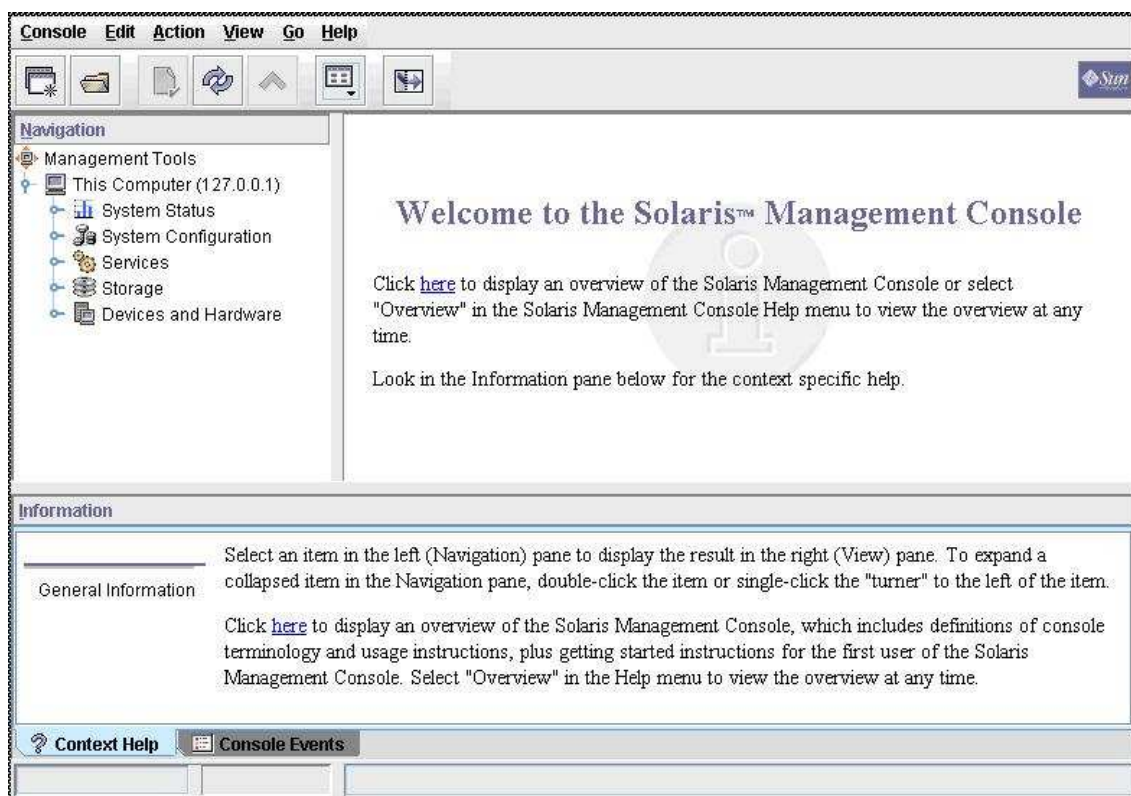
5 Administrace operačního systému

Správa operačního systému představuje činnosti vedoucí k bezproblémovému a očekávanému chodu operačního systému. Jedná např. o sledování kapacity souborového systému, spotřeby zdrojů výpočetního systému nebo změnu jeho nastavení.^[11]

Při výkonu administrátorských činností je správce do systému přihlášen jako privilegovaný uživatel root s UID=0. V tomto režimu je třeba dbát zvýšené opatrnosti, protože práva uživatele root umožňují se systémem provést cokoliv. Přihlášení pod uživatelem root je doporučeno omezit pouze na nezbytnou dobu.

K systémové administraci je možno použít grafické uživatelské rozhraní Solaris Management Console, prostřednictvím kterého uživatel komunikuje se Solaris Management Serverem a využívá běžné administrátorské utility. Nejúčinnějším

nástrojem však stále zůstává příkazový řádek. Jeho výhody oproti grafickému rozhraní tkví např. v možnosti automatizace běhu rutinních úkolů, snadné editaci konfiguračních souborů a existenci vnitřních příkazů, které jsou obdobou příkazů jazyka C a umožňují vytváření programů pro dávkové zpracování. Pro editaci konfiguračních souborů a psaní skriptů je běžně používán textový editor *vi*. Příkazový interpret je rovněž vhodný pro běh aplikací využívajících pouze textové rohraní.^[13]



Obrázek 1: Úvodní obrazovka Solaris Management Console verze 2.1

5.1 Instalace, balíčky a záplaty

5.1.1 Instalace operačního systému

Operační prostředí Solaris 10 je plně funkční na platformách SPARC a Intel. Pro procesory SPARC byl původně SunOS navržen. V následujícím textu bude soustředěna pozornost právě na tuto platformu. Platforma Intel podporovala na mikroprocesorech 80836 a 80846 SunOS 4.x. Sun Microsystems zvolil tuto platformu strategicky, neboť mnoho uživatelů osobních počítačů volí operační systémy typu UNIX, ale systémy s procesory SPARC pro ně představují stále finanční zátěž.

Specifický pro platformu SPARC je program Open Boot PROM, který slouží jako nástroj pro konfiguraci firmware, diagnostiku hardwarových komponent, instalaci a boot systému a testování konektivity. Je obdobou BIOSu na počítačích s architekturou procesoru x86.

Hardware, na který bude operační prostředí Solaris 10 instalováno, musí splňovat následující systémové nároky:

- minimálně 2 GB diskového prostoru (další požadavky diskové kapacity závisí na volbě softwarových balíčků nebo kapacitě swap oddílu určeném jako dvojnásobek velikosti operační paměti)
- minimální velikost operační paměti 256 MB, doporučených 512 MB, kdy pro tento parametr je určující množství aktivních uživatelů systému a počet plánovaných aplikací.
- platforma sun4u s 64-bitovou architekturou nebo x86 s 32 a 64-bitovou architekturou založenou na AMD, Intel a VIA x86 procesorech.^[26]

Provádí se buď kompletní instalace operačního systému či jeho upgrade. Celý proces je složen z předinstalace, samotné instalace a poinstalačních úprav. Kompletní instalaci je možno provést kdykoliv, pokud systém splňuje hardwarové nároky. Upgrade je možno provést v případě, že je nainstalován Solaris 7, 8, 9 nebo předchozí vydání Solaris 10 a za předpokladu splněných systémových požadavků. Pro instalaci lze zvolit jednu z instalačních metod. Metoda Web Start Wizard umožňuje přímou instalaci z lokální CD či DVD jednotky v grafickém rozhraní, zatímco suninstall používá příkazovou řádku. Má tedy nižší požadavky na systémové zdroje. Poslední instalační metodou je JumpStart, která umožňuje instalaci operačního systému z média sdíleného přes Network File System v lokální síti. Tato metoda se používá při instalaci většího počtu systémů. Požadované parametry instalovaných systémů lze nastavit v konfiguračním souboru *sysidcfg*.

5.1.2 Práce s balíčky

Instalace operačního systému se sestává ze softwarových balíčků a je dodávána na instalačním médiu. Balíček obsahuje soubory a adresáře se soubory spustitelnými, konfiguračními a s dokumentací.^[13] Balíčky jsou seskupovány do sad nazývaných

softwarové skupiny neboli clusteru. Jedna z fází instalace je výběr požadovaného clusteru.

Pro manipulaci se softwarovými balíčky slouží textové příkazy *pkgadd* k instalaci balíčku, *pkgrm* k jeho odstranění, *pkgchk* ke kontrole instalace. Příkaz *pkginfo* podává informace o balíčku, *pkgask* vytváří soubor pro dávkové zpracování instalace balíčku, *pkgparam* zobrazí hodnoty parametrů balíčku, *pkgtrans* přeloží jeho formát ^[16].

Tabulka 3: Použití příkazu *pkginfo* s přepínačem *-l* pro dlouhý výpis informací o softwarovém balíčku

```
root@sunray # pkginfo -l SUNWcsu
  PKGINST:  SUNWcsu
  NAME:     Core Solaris, (Usr)
CATEGORY:  system
  ARCH:    sparc
VERSION:   11.10.0,REV=2005.01.21.15.53
BASEDIR:   /
VENDOR:   Sun Microsystems, Inc.
  DESC:    core software for a specific instruction-set architecture
PSTAMP:   on10-patch20080629125923
INSTDATE: Aug 01 2008 19:36
HOTLINE:  Please contact your local service provider
STATUS:   partially installed
FILES:    1596 installed pathnames
          75 shared pathnames
          286 linked files
          138 directories
          466 executables
          30 setuid/setgid executables
          27740 blocks used (approx)
```

5.1.3 Záplaty

Záplaty neboli patche vznikají za účelem opravy chyb, které byly nahlášený vývojářskému týmu softwarového balíku. Jedná se např. o chyby bezpečnostního rázu. K modifikaci kódu dochází rovněž v případě urgentně vyžádané vlastnosti některé ze systémových aplikací.

V určitých intervalech jsou záplaty distribuovány k zákazníkovi v sadách na optickém datovém nosiči jako Enterprise Installation Services. Aplikace těchto sad záplat je doporučena po každé instalaci, aby byly ze systému odstraněny známé chyby. Rovněž je možné jednotlivé patche vyhledat a stáhnout z webových stránek Sun Microsystems na internetové adrese *sunsolve.sun.com*. Jejich identifikace probíhá

na základě šestimístného numerického kódu v názvu patche, který je doplněn o dvojčíslí číslo revize za pomlčkou.

K získávání informací o instalovaných záplatách slouží příkaz *showrev -p*. Ve výstupu je číslo záplaty, dále čísla záplat, jejichž platnost byla touto záplatou zrušena, čísla záplat, kterými je podmíněna instalace nové záplaty, čísla nekompatibilních záplat a názvy balíčků, na které má tato záplata vliv. Zkrácený příklad výstupu příkazu ilustruje Tabulka 4: Výstup příkazu *showrev -p*.

Tabulka 4: Výstup příkazu *showrev -p*

```

root@sunray # showrev -p |head
Patch: 118367-04 Obsoletes: Requires: Incompatibles: Packages:
SUNWcsu
Patch: 118872-04 Obsoletes: Requires: Incompatibles: Packages:
SUNWcsu
Patch: 118918-24 Obsoletes: 116781-02, 119012-03, 121282-02, 121284-
02, 121292-01, 121473-01, 121476-01, 121478-01, 121786-01, 123444-01
Requires: Incompatibles: Packages: SUNWcsu, SUNWcsr, SUNWcsl,
SUNWcnetr, SUNWcar, SUNWcakr, SUNWckr, SUNWcslr, SUNWhea, SUNWmdb,
SUNWmdbr
...
(výstup zkrácen)

```

Pokud je požadován přehled o všech záplatách aplikovaných na konkrétní softwarový balík, jejich seznam je výstupem příkazu *pkgparam <JMÉNO_BALÍKU> PATCHLIST*. Údaj o datu aplikace záplaty poskytne příkaz *pkgparam <JMÉNO_BALÍKU> PATCH INFO <ID_ZÁPLATY>*.

Pro instalaci záplat do systému slouží příkaz *patchadd <ID_ZÁPLATY>*, pro odstranění pak *patchrm <ID_ZÁPLATY>* a dále nový nástroj *smpatch* pro stahování, aplikací a odstraňování aktualizací.^[17]

5.2 Spuštění a zastavení operačního systému

V této kapitole je popsán start a zastavení operačního systému, který je rovněž pro platformu SPARC specifický.

5.2.1 Zapnutí a vypnutí operačního systému

V závislosti na nastavení proměnné prostředí Open Boot PROM *auto-boot?* je operační systém spuštěn buď automaticky po zapnutí výpočetního systému nebo ručně z prostředí Open Boot PROM příkazem *boot*. Volitelné parametry příkazu *boot* určují

zařízení pro zavedení operačního systému, kterým může být buď pevný disk, optické médium, páska nebo lokální síť. Použitím přepínačů lze z Open Boot PROM nabootovat rovnou do jednovivatelského režimu (*boot -s*), zvolit alternativní parametry zavedení operačního systému (*boot -a*) nebo při odebrání či přidání hardwarových komponent výpočetního systému provést rekonfigurační boot (*boot -r*).

Tabulka 5: Pracovní stanice sunray má povoleno automatické zavedení operačního systému, server macek nikoliv

```
root@sunray / # eeprom auto-boot?
auto-boot?=true
#####
root@macek ~ # eeprom auto-boot?
auto-boot?=false
```

Operační prostředí Solaris 10 je určeno pro nepřetržitý provoz. Tato skutečnost se opírá o redundantní a vysoce dostupnou konfiguraci technických prostředků provozních výpočetních systémů. Jde zde především o redundantní napájecí zdroje, pevné disky uspořádané do RAID-1 s možností výměny za provozu systému apod. Přes všechny tyto vlastnosti je někdy nutné operační systém zastavit. K tomuto účelu lze použít dále uvedené příkazy.

Přepínač *-i* příkazu *shutdown* podobně jako příkaz *init* změní provozní režim operačního systému. Pomocí přepínače *-g* lze navíc specifikovat délku prodlevy, která slouží pro ukončení všech kritických procesů. U tohoto příkazu je možné pomocí přepínače *-y* taktéž potvrdit zamýšlené zastavení operačního systému.

poweroff slouží pro okamžité zastavení a vypnutí systému, kdy nejsou vykonávány ukončovací skripty v jednotlivých režimech provozu. Tento způsob je určen pro použití v havarijních případech, jako je výpadek elektrické energie apod.

halt je obdobou příkazu *poweroff*. Má také stejné přepínače.

Před zastavením operačního systému je doporučena a v praxi ověřena synchronizace souborových systémů příkazem *sync*. Dochází tak k zápisu obsahu vyrovnávací paměti na disk.^[13]

5.2.2 Boot proces

Boot systému je procesem zavedení jádra operačního systému do paměti počítače. Pro hardware založený na platformě SPARC se tento proces liší díky specifickému firmwaru.

Po zapnutí počítače je systém uveden do režimu Open Boot PROM, ve kterém program *monitor* provede diagnostiku paměti, disků, vstupně výstupních zařízení apod. Po jejím úspěšném ukončení je ze zařízení určeného pro zavedení systému načten primární zavaděč *bootblk*. Je lokalizován v prvních 15 sektorech disku. Primární zavaděč lokalizuje a ze souborového systému UFS zavede program *ufsboot*, který se postará o zavedení obou částí jádra. Po své inicializaci jádro načítá moduly a spustí proces *init* s PID=1.^[3]

5.2.3 Service Management Facility (SMF)

V předcházejících verzích operačního systému proces *init* po spuštění četl soubor */etc/inittab*, v němž byly instrukce ke spouštění dalších procesů. S příchodem Solaris 10 byl obsah tohoto souboru značně zredukován. Zůstal v něm však záznam pro proces *svc.startd*, který převzal zodpovědnost za start a vypínání procesů. V jeho režii je i přechod mezi jednotlivými provozními režimy operačního systému.

Proces *svc.startd* je primárním procesem, který zajišťuje chod služeb operačního systému v rámci infrastruktury Service Management Facility. V tomto modelu je automaticky zajištěn opětovný start zhavarovaných služeb s ohledem na všechny závislosti startovaných procesů. Pomocí zřetelných závislostí je možné snáze najít příčinu havárie služby. Konfigurace služeb může být prostřednictvím automatických snímků zálohována a podle potřeby obnovena. Pro modifikaci nebo restart systémových služeb je možné provést delegaci na neprivilegovaného uživatele.

Start a zastavení operačního systému Solaris 10 je rychlejší než v předchozích verzích, neboť podle závislostí definovaných v SMF služby mohou startovat souběžně. Rovněž výstup na konzoli je při startu systému standardně stručnější. Veškeré zprávy o startu operačního systému jsou ukládány do */var/svc/log*. Ten lze ovšem pomocí parametrů příkazu *boot* učinit podrobnější.

Přestože řada tzv. init skriptů užívaných v předchozích verzích operačního systému pro zabezpečení chodu služeb byla z adresářů `/etc/init.d` a `/etc/rc*.d` – stejně tak jako záznamy v tabulce `/etc/inittab` – odstraněna, Service Management Facility zůstává s jejich mechanismem plně kompatibilní. Zbývající služby, jež jsou řízeny init skripty, jsou startovány mezi posledními, takže jejich závislosti jsou splněny.

V rámci SMF je základní správním jednotkou instance služby. Každá instance je specifickou konfigurací služby a může být nezávisle nastartována či zastavena. V systému může běžet více instancí téže služby. Služby jsou reprezentovány objekty a jejich instance jsou potomky těchto objektů. Každá služba v systému je identifikována tzv. FMRI (Fault Management Resource Identifier). Služby řízené init skripty jsou od ostatních svým FMRI odlišitelné. Definice vlastností služeb nebo jejich instancí jsou uloženy v souborech formátu XML v adresáři `/var/svc/manifest`. Není doporučeno je editovat manuálně, protože pro konfiguraci služeb slouží příkazy `svcadm`, `svcs`, `svccfg`.

Tabulka 6: Příklad služeb řízených init skripty; zakázaných a běžících v rámci SMF

```

root@sunray / # svcs -a |more
STATE          STIME          FMRI
legacy_run     Mar_13         lrc:/etc/rc2_d/S31lutsyscfg
legacy_run     Mar_13         lrc:/etc/rc2_d/S401lc2
... (výpis přerušen)
disabled       Mar_13         svc:/network/talk:default
disabled       Apr_02         svc:/network/rpc/wall:default
online         Mar_13         svc:/system/svc/restarter:default
online         Mar_13         svc:/network/pfil:default
online         Mar_13         svc:/network/loopback:default
online         Mar_13         svc:/system/filesystem/root:default
    
```

Skupina určitých služeb definuje provozní režim. V SMF je označována jako *milestone* a koresponduje s dříve užívanými *run levels*, na které se odkazuje např. příkaz `who -r`. Definované provozní režimy slouží k různým účelům, např. ke správě systému v jednouchyvatelském režimu.^[3]

Tabulka 7: Výpis běžících služebních skupin

```

root@sunray / # svcs -a |grep milestone
online         Mar_13         svc:/milestone/devices:default
online         Mar_13         svc:/milestone/single-user:default
online         Mar_13         svc:/milestone/name-services:default
online         Mar_13         svc:/milestone/sysconfig:default
online         Mar_13         svc:/milestone/multi-user:default
online         Mar_13         svc:/milestone/multi-user-server:default
online         Apr_02         svc:/milestone/network:default
    
```

Tabulka 8: Výpis aktuálního provozního režimu

```
root@sunray / # who -r
.          run-level 3  Mar 13 14:21      3      0  S
```

5.3 Správa disků, souborové systémy a správa datových svazků

Při každém startu operačního systému je vytvořena hierarchická struktura všech hardwarových komponent. Tato struktura je pod kontrolou souborového systému `devfs` a je umístěna do adresáře `/devices`. Každý soubor v tomto adresáři je logickou reprezentací jedné hardwarové komponenty. Jména těchto souborů jsou totožná s fyzickými názvy zařízení. Protože tyto soubory nemají žádné datové bloky, místo údaje o velikosti jejich i-uzly obsahují informaci o tzv. MAJOR a MINOR číslech. MAJOR číslo určuje příslušný ovladač zařízení v jádře a MINOR slouží pro identifikaci zařízení v rámci jednoho ovladače.^[4]

Administrátorovi systému a některým uživatelským aplikacím slouží pro identifikaci systému tzv. logické jméno zařízení. Všechna tato jména jsou umístěna ve struktuře adresáře `/dev` jako symbolické odkazy na speciální soubory.

Solaris 10 podporuje na platformě SPARC dynamickou rekonfiguraci disků a některých dalších zařízení. Tato vlastnost dovoluje, aby za běhu systému byla přidána nová nebo nahrazena nefunkční hardwarová komponenta a bez rebootu systému obnoven logický strom zařízení. Pomocí příkazu `devfsadm` a jeho přepínačů je dosaženo propagace nových zařízení do operačního systému a je aktualizována logická struktura zařízení.

5.3.1 Nástroj *format*

Tento nástroj je určen pro správu diskových oddílů. Byl původně vyvinut pro správu SCSI disků. Jsou-li dostupné určité ovladače zařízení, je možné jej použít i pro správu optických disků nebo logických jednotek vypropagovaných z diskových polí. Pro práci s příkazem *format* je potřeba orientovat se v následujících termínech.

Jmenovka disku je speciální prostor obsahující informace o disku jako je jeho geometrie a oddíly. Je umístěn v prvním sektoru disku. V operačním systému je označován jako *Volume Table Of Contents*, zkráceně VTOC.

Diskový oddíl je vytvořen ze sousedících bloků na disku a nemůže přesahovat rámec jednoho fyzického disku. Na každém diskovém oddíle může být vytvořen nejvíce jeden souborový systém a ten nemůže překročit jeho rámec.

Seznam defektů zaznamenává všechny oblasti disku, které nemohou být použity pro zápis a čtení dat. S postupem času tento seznam roste.

Tabulka oddílů je součástí VTOC a obsahuje seznam oddílů, hranice oddílu a jeho velikost.^[10]

Pomocí nabídek příkazu *format* lze disk zformátovat, opatřit jmenovkou, opravit nebo analyzovat. Po zadání příkazu se objeví nabídka disků, které byly přidány do systému buď rekonfiguračním bootem nebo dynamicky příkazem *devfsadm*.

Tabulka 9: Nabídka interaktivního příkazu "format"

```
root@macek ~ # format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
  0. c0t0d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
    /pci@1f,4000/scsi@3/sd@0,0
  1. c0t1d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
  2. clt1d0 <FUJITSU-MAG3182LSUN18G-1111 cyl 7506 alt 2 hd 19 sec
248>
    /pci@1f,4000/scsi@3,1/sd@1,0
  3. clt2d0 <SEAGATE-ST318406LC-010A cyl 26123 alt 2 hd 2 sec 686>
    /pci@1f,4000/scsi@3,1/sd@2,0
  4. clt3d0 <COMPAQ-BB00911CA0-3B07 cyl 5271 alt 2 hd 20 sec 168>
    /pci@1f,4000/scsi@3,1/sd@3,0
Specify disk (enter its number): 4
```

Volbou příslušného čísla disku je aktivována nabídka činností, které lze nad diskem provádět.

Tabulka 10: Nabídka úkolů, které lze nad diskem provést pomocí interaktivního příkazu "format"

```
selecting clt3d0
[disk formatted]

FORMAT MENU:
  disk      - select a disk
  type      - select (define) a disk type
  partition - select (define) a partition table
  current   - describe the current disk
  format    - format and analyze the disk
  repair    - repair a defective sector
  label     - write label to the disk
  analyze   - surface analysis
  defect    - defect list management
  backup    - search for backup labels
  verify    - read and display labels
  save      - save new disk/partition definitions
  inquiry   - show vendor, product and revision
  volname   - set 8-character volume name
  !<cmd>    - execute <cmd>, then return
  quit
```

5.3.2 Práce se souborovými systémy

Pro účely práce s daty jsou na diskových oddílech vytvářeny souborové systémy. Standardem pro UNIXové operační systémy se stal Unix File System, UFS. K jeho založení slouží interaktivní příkaz *newfs*.

Velmi pružným souborovým systémem je Zettabyte File System, zkráceně ZFS. Tento souborový systém byl vytvořen společností Sun Microsystems. Uvolnění pod open-source licencí CDDL umožnilo jeho port na další operační systémy. Dnes je součástí např. UNIXových distribucí založených na BSD a operačního systému Mac OS X.

Síla ZFS tkví v možnosti vytváření souborových systémů téměř neomezené velikosti. ZFS již nepotřebuje diskové oddíly. Namísto toho, kapacita dostupných disků je soustředěna v tzv. *pool*, z něhož je vytvořena vlastní hierarchie souborových systémů. Výhodou je i podpora neomezeně velikých souborů.^[22]

Kromě UFS a ZFS jsou v Solaris 10 dostupné i další diskové souborové systémy jako je HSFS používaný na kompaktních discích a určený pouze pro čtení, UDF určený pro optické DVD disky. Do operačního systému je rovněž možné připojit souborové systémy typu NFS, které jsou zpřístupněny přes síť z ostatních serverů.

Poslední skupinou jsou virtuální souborové systémy, např. již zmiňovaný souborový systém DEVFS, které umožňují přístup k systémovým informacím jádra. Např. PROCFS obsahuje aktivní procesy systému a je připojen v adresáři */proc*. TMPFS je standardním souborovým systémem pro adresář */tmp*, který slouží pro dočasné soubory. Většinou jsou vytvořeny v paměti, takže nevyužívají diskovou kapacitu. Výjimkou je SWAPFS určený k odkládání procesů jádrem na disk při nedostatečné kapacitě operační paměti.^[3]

Tabulka */etc/vfstab* popisuje standardní způsob připojení konfigurovaných souborových systémů při bootu. Jednotlivé položky zobrazují zleva doprava blokové zařízení připojované do hierarchie kořenového adresáře, znakové zařízení využívané nástrojem fsck pro kontrolu konzistence dat, standardní adresář připojení, typ souborového systému, přepínač pro automatickou kontrolu konzistence dat, volbu automatického připojení souborového systému a vlastnosti připojení.

Tabulka 11: Tabulka */etc/vfstab* popisující vlastnosti souborových systémů

5.3.3 Správa datových svazků

Servery s operačním systémem Solaris jsou konfigurovány tak, aby poskytovaly stabilní a vysoce dostupné prostředí pro běh databází, aplikačních serverů a distribuovaných aplikací často kladoucích vysoké nároky na kapacitu diskových úložišť. Solaris Volume Manager v předchozích verzích operačního systému označovaný jako Soltice DiskSuite je software pro správu datových svazků. Umožňuje vytváření vysokokapacitních virtuálních disků, tzv. svazků, zvýšení datové dostupnosti a usnadňuje správu velkého množství disků. Pomocí tohoto software lze v některých případech dosáhnout i zlepšení výkonu vstupně výstupních operací. Operační systém

poskytuje podporu pro standard RAID. V RAID je nadefinováno několik typů umožňujících např. ukládání dat jednoho datového svazku s prokládáním na více disků (*striping*), nebo rozkopírování datových svazků do více instancí (*mirroring*). Striping umožňuje právě zvýšení výkonu vstupně výstupních operací a mirroring poskytuje datovou redundanci, čímž zvyšuje datovou dostupnost a spolehlivost systému.

Z aplikačního hlediska je svazek ekvivalentem fyzického disku. Je také označován jako virtuální či meta zařízení. Práce s disky je transparentní, takže vstupně výstupní operace vykonávané operačním systémem na datovém svazku jsou softwarem směrovány na jednotlivé disky.

Pro správu datových svazků v programu Solaris Volume Manager jsou nutná práva superuživatele a lze využít grafické rozhraní Solaris Management Console nebo sadu příkazů textového rozhraní, mezi které patří např. *metainit*, *metastat*.^[21]

5.3.4 Zálohování a obnova dat

Zálohování zajišťuje zvýšenou dostupnost dat a patří mezi klíčové činnosti správce operačního systému v prostředích s kritickými daty. Zálohy jsou používány k obnovám zničených dat. Doporučuje se je vykonávat pravidelně. Následující text poskytuje výčet systémových nástrojů pro zálohování.

Pro kopírování souborových systémů nebo jednotlivých souborů slouží nástroj *tar*. Umožňuje také vytváření resp. rozbalování archivů na znaková resp. ze znakových zařízení.

Vhodným nástrojem pro kopírování identických disků je *dd*. Tento nástroj pracuje se znakovými zařízeními, je tedy rychlý a umožňuje konverzi blokových velikostí. Kopíruje po blocích vstupní soubor do výstupního souboru a implicitně využívá standardní vstup a výstup. Po dokončení operace je poskytnut výstup o množství zkopírovaných bloků.

Malé archívy díky jejich jednoduché hlavičce vytváří nástroj *cpio*, který umožňuje i kopírování jednotlivých souborů a také uložení jediného archivu na více datových nosičů. Utilita pracuje ve třech režimech a je nutné jeden z nich explicitně zadat.

cpio -o vytvoří zálohu na určeném datovém nosiči

cpio -i slouží pro extrakci záložního archivu

cpio -p je používáno při kopírování souborů z disku na disk

Nástroj *ufsdump* je speciálně navržen pro zálohování souborových systémů. Je ovšem použitelný pouze pro neaktivní souborové systémy, tzn. nepřipojené nebo v režimu pouze pro čtení. Pokud je kopírovaný souborový systém aktivní, data ve vyrovnávací paměti se nezkopírují a zálohovaná data mohou být nekonzistentní. Pro obnovu záloh vytvořených nástrojem *ufsdump* slouží příkaz *ufsrestore*.

Kombinace posledních dvou uvedených nástrojů je vhodná pro klonování disků, kdy je nejprve třeba zkopírovat tabulku oddílů a pak pomocí *ufsdump* a *ufsrestore* zkopírovat jednotlivé diskové oddíly.

5.4 Správa uživatelů

5.4.1 Uživatelé a vytváření uživatelských účtů

Správa uživatelů je v operačním systému typu UNIX nutností, protože žádný proces ani soubor nemůže existovat, aniž by měl definovaného uživatele a jeho skupinu.^[13] Všichni uživatelé v systému mají přidělený jednoznačný numerický identifikátor, UID, a další atributy.

Za předpokladu, že není používána žádná adresářová služba jako např. NIS či LDAP, informace o uživatelských účtech jsou uloženy v tabulkách uložených v souborech adresáře *etc/*. Pokud jsou adresářové služby používány, lokální konfigurační soubory uživatelského přístupu v systému přesto existují. Těmito klíčovými soubory pro správu uživatelů jsou */etc/passwd*, */etc/shadow* a */etc/group*. V tabulce */etc/shadow* jsou nejdříve uvedeny systémové účty vytvořené pro běh služeb systému a teprve pak následují účty uživatelské. V jednotlivých položkách v tabulce jsou uvedeny zleva doprava uživatelské jméno, příznak hesla, ID uživatele, ID skupiny uživatele, poznámka k uživatelskému účtu, domovský adresář a standardní příkazový interpret. Položka „příznak hesla“ v souboru */etc/passwd* se odkazuje na zakrytované heslo, které je ovšem z bezpečnostních důvodů uloženo v souboru */etc/shadow*. Přístup do tohoto souboru má pouze superuživatel. Položky souboru zleva doprava obsahují uživatelské jméno, zakrytované heslo uživatele, počet dní mezi poslední změnou hesla

a 1. lednem 1970, minimální počet dní, po kterém může dojít ke změně hesla, maximální počet dní platnosti hesla, než je uživatel vyzván k jeho změně, a dobu ve dnech, kdy účet zůstane nepoužívaný předtím, než je uzamčen. Poslední položkou je datum vypršení platnosti účtu.

Tabulka 12: Informace o uživatelských účtech v /etc/passwd

```
root@sunray / # cat /etc/passwd
root:x:0:0:Super-User:/:/sbin/sh
daemon:x:1:1:/:
bin:x:2:2:/:usr/bin:
sys:x:3:3:/:
adm:x:4:4:Admin:/var/adm:
... (výpis přerušen)
polj1480:x:2011:10:Jan Polak:/export/home/polj1480:/bin/bash
maco1575:x:2012:10:Ondrej Mach:/export/home/maco1575:/bin/bash
dalj1674:x:2013:10:Jiri Dalik:/export/home/dalj1674:/bin/bash
```

Vytváření uživatelských účtů je možné provádět buď v grafickém prostředí Solaris Management Console nebo z příkazové řádky. V textovém rozhraní je možné určit víc atributů uživatelského účtu než v grafickém prostředí. Pro práci s uživatelskými účty má Solaris 10 příkaz *useradd*. Následující příklad ilustruje vytvoření uživatele *tmpusr*, který bude mít z příkazové řádky určeno uživatelské ID 777, skupinové ID 10, domovský adresář */export/home/user1*, příkazový interpret */bin/bash*, poznámku „temporary user“, datum expirace 9.9.2009, jméno přiřazeného projektu *bigbang*. Přepínač *-m* příkazu *useradd* znamená automatické založení domovského adresáře, pokud neexistuje.

Tabulka 13: Vytvoření nového uživatelského účtu

```
root@sunray / # useradd -u 777 -g 10 -c "temporary user" -d
/export/home/tmpusr -m \ -s /bin/bash -p bigbang tmpusr
```

Vytvoření nového uživatele se promítne v souborech */etc/passwd*, */etc/shadow* a */etc/projects*.

Tabulka 14: Založení nového uživatele a záznamy v příslušných souborech

```
root@sunray / # for i in /etc/passwd /etc/shadow /etc/project; do echo
"";echo $i;;grep tmpusr $i; done

/etc/passwd:
tmpusr:x:777:10:temporary user:/export/home/tmpusr:/bin/bash

/etc/shadow:
tmpusr:*LK*::::::14496:
```

```
/etc/project:  
bigbang:100::tmpusr::
```

Není doporučeno konfigurační soubory v adresáři */etc* upravovat ručně. Pokud je nevyhnutelné editovat soubor */etc/passwd* ručně, je třeba použít příkaz *pwck*, který zaručuje konzistenci dat tohoto souboru.

5.4.2 Bezpečnost

Kromě ochrany dat proti ztrátě v důsledku selhání systémových komponent je třeba operační systém zabezpečit proti přístupu neoprávněných uživatelů. Zatímco ztrátě kritických dat v důsledku havárie operačního systému lze předcházet pravidelným zálohováním, ochranu proti zneužití přístupu k systému lze zajistit na několika úrovních, a to:

- Kontrolou fyzického přístupu k hardware
- Kontrolou přístupu do systému pomocí hesel a uživatelských oprávnění
- Kontrolou přístupu k souborům
- Kontrolou uživatelské aktivity
- Kontrolou přístupu po síti
- Zabezpečením přístupu administrátora.^[3]

Vztah počtu úrovní zabezpečení a možností neoprávněného přístupu má záporný korelační koeficient.

Pokud se jedná o fyzický přístup, kritický je jak přístup k systémové konzoli, tak k datovým zálohám. Co se týká kontroly uživatelských přístupů, je pozornost soustředěna na používání přístupových hesel a práv. Každý uživatelský účet by měl být opatřen heslem. Bezpečnostní politiky pro uživatelská hesla a přístupy do systému lze na úrovni operačního systému nastavit v adresáři */etc/default/* v souborech *passwd*, *login* a *su*.^[3]

Soubory a adresáře jsou v operačním systému zabezpečeny pomocí přístupových práv. Schéma přístupových práv je společné všem operačním systémům na bázi UNIX. Jedná se o sadu práv ke čtení, zápisu a spouštění spustitelných souborů pro vlastníka, skupinu vlastníka a ostatní uživatele. Nastavení přístupových práv, resp. vlastníka

souboru, resp. skupiny vlastníka je docíleno pomocí příkazů *chmod*, resp. *chown*, resp. *chgroup*.

5.4.3 Návrh skriptu pro hromadné vytváření uživatelů

V této kapitole je uveden příklad skriptu pro dávkové zpracování úlohy napsaného v příkazovém interpretu Bash.

Pro tento vzorový příklad platí, že systém nevyužívá adresářové služby. Je však třeba vytvořit více uživatelů najednou. Takováto rutinní činnost může být právě pomocí skriptu příkazového interpretu zautomatizována.

Jako vstup skriptu slouží textový soubor se seznamem požadovaných uživatelů. Platí předpoklad, že soubor je naplněn daty ve správném formátu, tj. v tomto případě jsou jednotlivé atributy odděleny čárkou. Položka v seznamu má atributy křestní jméno, příjmení a ročník narození.

Ve skriptu jsou definovány lokální proměnné:

USER_LIST: soubor se seznamem uživatelů

USER_LIST_H: pomocný seznam uživatelů

a dále jsou používány proměnné:

n: pořadová proměnná v seznamu uživatelů

i: proměnná sloužící pro iteraci v generátoru loginů

first_last_vol: pole atributů jednotlivých položek

last_vol: pole vybraných atributů jednotlivých položek

FIRST: proměnná pro atribut křestní jméno

LAST: proměnná pro atribut příjmení

VOL: proměnná pro atribut ročník narození

F: pomocná proměnná pro iniciálu křestního jména

L: pomocná proměnná pro první tři znaky příjmení

DATE: proměnná pro rok započetí studia

UNCD: kód zaručující unikátnost loginů (součet hodnot *n* a *DATE*)

LOGIN: výsledný login uživatele

Prvním krokem skriptu je kontrola existence souboru uživatelů. Pokud soubor neexistuje, skript skončí s návratovým kódem 1. Následuje kontrola pomocného

souboru se seznamem uživatelských přihlašovacích jmen. Pokud soubor existuje, je smazán.

Pořadová proměnná seznamu uživatelů n je naplněna hodnotou 0. Proměnná $DATE$ je naplněna hodnotou aktuálního data ve formátu posledních dvou číslic letopočtu. Dále následuje iterace pro jednotlivé položky seznamu uživatelů, ve které je:

1. Proměnná $first_last_vol$ naplněna položkou ze seznamu uživatelů $USER_LIST$,
2. řetězcem vzniklým odstraněním nejkratšího vyhovujícího řetězce před vzorkem „,“ z proměnné $first_last_vol$ naplněna proměnná $last_vol$,
3. řetězcem vzniklým odstraněním nejdelšího vyhovujícího řetězce za vzorkem „,“ z proměnné $first_last_vol$ naplněna proměnná $FIRST$,
4. prvním znakem z proměnné $FIRST$ převedeným z velkého na malé písmo naplněna proměnná F ,
5. řetězcem vzniklým odstraněním nejdelšího vyhovujícího řetězce za vzorkem „,“ z proměnné $last_vol$ je naplněna proměnná $LAST$,
6. proměnná L je naplněna prvními třemi znaky, které jsou převedeny z velkého na malé písmo,
7. řetězcem vzniklým odstraněním nejkratšího vyhovujícího řetězce před vzorkem „,“ z proměnné $last_vol$ je naplněna proměnná VOL
8. je inkrementována proměnná n ,
9. hodnotou vzniklou aritmetickým součtem hodnot n a $DATE$ naplněna proměnná $UNCD$,
10. proměnná $LOGIN$ naplněna hodnotou vzniklou zřetěžením hodnot L , F a $UNCD$ (loginem uživatele)
11. do pomocného seznamu $USER_LIST_H$ přidána položka ve formátu „ $LAST$, $FIRST$, $LOGIN$ “,
12. vytvořen uživatelský účet pro hodnotu proměnné $LOGIN$, se standardním příkazovým interpretem $/bin/bash$, domovským adresářem označeným podle loginu uživatele v adresáři $/export/home$, skupinou uživatele „staff“ komentářem se jménem uživatele,

13. pro uživatelský účet je vymazáno heslo a při prvním přihlášení uživatele je vynuceno zadání nového hesla.

Výstupem tohoto skriptu jsou založené uživatelské účty dle seznamu požadovaných uživatelů.

Příloha č. 1 - skript příkazového interpretu bash pro dávkové vytvoření uživatelských účtů ze seznamu uživatelů obsahuje přepis skriptu.

5.5 Sítě

V dnešních počítačových sítích je využíván k přenosu dat čtyřvrstvý model rodiny protokolů TCP/IP. Protokoly určují pravidla, podle kterých v jednotlivých vrstvách síťové komunikace fungují hardwarové i softwarové komponenty a zpracovávají posílaná data.

Standardem linkové vrstvy je Ethernet. Definuje pravidla pro fyzické komponenty používané v síťové komunikaci. Jeho součástí jsou specifikace pro konektory, kabely a síťové karty.^[3] Každá síťová karta je jednoznačně identifikovatelná svou hardwarovou adresou neboli MAC adresou.

Podle standardu Internet Protocol verze 4 definovaném v síťové vrstvě modelu TCP/IP má každá síťová karta 32-bitovou adresu neboli IP adresu. IP adresy se udávají v oktálovém zápisu a bajty jsou odděleny tečkou. V rámci sítě nemohou fungovat dvě stejné IP adresy.

V sítích je propojeno velké množství počítačů pomocí aktivních síťových prvků switchů a hubů. Jendotlivé sítě jsou mezi sebou propojeny pomocí počítačů s více síťovými kartami. Data jsou mezi jednotlivými sítěmi směrována prostředky routerů.

Díky nově navržené sadě protokolů síťové vrstvy nabízí Solaris 10 znamenité výkonnostní a bezpečnostní vlastnosti síťové komunikace. Důležitá je i podpora široké škály síťového hardware a technologií, která ze Solaris 10 činí platformu vhodnou pro provoz síťových služeb a aplikací.^[4]

5.5.1 Konfigurace síťových rozhraní

Pro konfiguraci síťových rozhraní je určen příkaz *ifconfig* a jeho přepínače. Umožňuje přiřazení IP adresy síťové kartě a konfiguraci parametrů síťové karty. Tento

příkaz je součástí init skriptů, takže po bootu počítače jsou na síťových kartách nakonfigurovány IP adresy. Dále se příkaz používá pro změnu IP adresy a dalších parametrů jako je maska podsítě a všesměrová adresa, vypnutí a zapnutí síťového rozhraní. Pokud je v příkazové řádce použit příkaz bez přepínačů, výstupem je informace o aktivních síťových kartách. Přehled všech konfigurovaných rozhraní je možné zobrazit pomocí přepínače *-a*. Měnit konfiguraci síťových karet může pouze superuživatel. V operačním systému jsou dvě verze příkazu, které jsou volány s ohledem na fakt, zda-li systém používá jmenné služby. Důležitým parametrem příkazu je volba *plumb*, která otevírá zařízení spojené se jménem síťové karty a nastavuje jej pro použití protokolem IP.^[17]

Tabulka 15: Zobrazení všech dostupných síťových rozhraní v systému

```
root@macek ~ # ifconfig -a
lo0: flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu
8232 index 1
    inet 127.0.0.1 netmask ff000000
hme0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500
index 2
    inet 172.16.0.38 netmask ffff0000 broadcast 172.16.255.255
    ether 0:3:ba:1d:3f:43
```

Dalším příkazem umožňujícím konfiguraci síťových rozhraní je příkaz *ndd*. Je používán k nastavování parametrů přenosu protokolu TCP/IP přímo v ovladačích v jádře.

5.5.2 Jmenné a adresářové služby

Jmenné služby udržují centrální úložiště dat, jehož informace jsou používány uživateli, počítači a aplikacemi ke vzájemné komunikaci v síti. V prostředí bez centrální jmenné služby jsou k tomuto účelu určeny lokální konfigurační soubory v adresáři */etc*. Adresářová služba jako server přijímá žádosti od klientů a poskytuje níže uvedené informace:

- jména počítačů
- uživatelská jména
- hesla
- přístupová práva.

Kromě správy informací v centrálním úložišti dat jmenné služby rovněž usnadňují komunikaci v síti. Poskytují totiž vazbu objektu na jméno, kterým je objekt v síti identifikován.

V prostředí operačního systému Solaris je možné pracovat s několika typy adresářových služeb. Synchronizaci použití těchto služeb určuje obsah konfiguračního souboru `/etc/`.

Služba *Domain Name Service* slouží pro překlad IP adres počítačů na jména a obráceně jak v lokální administrátorské doméně, tak v rámci celého internetu. Jednotlivé DNS domény jsou skupiny počítačů a jsou hierarchicky uspořádány. Celek tvoří tzv. jmenný prostor. V každé doméně musí být jeden primární jmenný server a nejméně jeden záložní. DNS služba je zajišťována procesem *in.named*. Na straně klienta je DNS implementována tzv. resolverem, který žádá jmenný server o poskytnutí informací. Jmenný server buď vrací požadovanou informaci, nebo odkazuje na ostatní jmenné servery.

Pro zjednodušení správy lokální sítě je určena služba *Network Information Service*. O souboru informací, které tato služba spravuje a poskytuje, se hovoří jako o jmenném prostoru. Jeho data jsou uložena v tzv. jmenných mapách, které nahrazují lokální konfigurační soubory. Podobně jako DNS je NIS klient-server aplikace. Hierarchicky uspořádané servery poskytují informace klientům.

Služba NIS Plus je obdobou NIS, nicméně má mnohem více funkčních vlastností. Je to dynamická služba, protože informace mohou být kdykoliv aktualizovány. Hierarchická struktura jmenného prostoru umožňuje vytvoření obrazu logického uspořádání podniku. V NIS Plus je integrovaný bezpečnostní systém pro ochranu jmenného prostoru služby a informací v něm uložených. Pro uživatele služby je použita autentifikace k ověření samotného uživatele sítě a utorizace k ověření uživatelských práv na informaci.^[24]

Lightweight Directory Access Protocol je adresářová služba pro správu informací o uživateli a zdrojích v rozlehlé síti rozdělené do více domén. Podobně jako NIS Plus a DNS poskytuje hierarchickou strukturu informací. Součástí služby je i autentifikační mechanismus pro ověřování žadatelů. Služba LDAP má velký potenciál a pomalu vytlačuje NIS a NIS Plus z prostředí Solaris 10.^[3]

6 Výkon a jeho monitorování

U některých operačních systémů nové vydání znamená řadu nových funkcí a vlastností, které ovšem mohou představovat velkou zátěž na systémové zdroje. Někdy bývá nutností s aktualizací operačního systému provést i pořízení nových výkonnějších technických prostředků. Naštěstí to není případ operačního systému Solaris 10, který oproti předchozím verzím zaznamenal značné zvýšení výkonu.^[8]

6.1 Sledování výkonu

6.1.1 *iostat*, *vmstat*, *netstat*

Nejpoužívanějšími nástroji pro sledování výkonu jsou integrované utility *iostat*, *vmstat*, *netstat*.

Nástrojem *iostat* lze vytvářet statistiky o vstupně výstupních operacích nad disky nebo terminálem. Také lze získat přehled o využití procesorů. V prvním řádku výstupu tohoto příkazu je sumární přehled aktivity od naboování systému. Každý další řádek podává informaci o bezprostředně předcházejícím časovém intervalu, který lze určit jako parametr v příkazovém řádku. Pro každé vstupně výstupní zařízení jádro počítá čtení, zápisy, přečtené bajty, zapsané bajty. Výstup utility je nanejvýš přesným měřítkem rychlosti transakcí, propustnosti sběrnic, využití zařízení nebo procesorového času spotřebovaného pro transakci. Pro účely statistik jádro vytváří počítadla.^[17]

Nástroj *vmstat* poskytuje statistiky o procesech nebo vláknech jádra, virtuální paměti, discích a aktivitě procesoru. Číslo v prvním sloupci výstupu příkazu *vmstat* udává počet procesů čekajících na procesor. Pokud je toto číslo menší než čtyřnásobek počtu procesorů, lze hovořit o optimální zátěži systému. Údaje posledního sloupce výstupu příkazu podávají informace o čase, který procesor stráví v nečinnosti. Pokud jsou tato čísla rovna nule a čas v režimu jádra je dvojnásobkem času v režimu uživatele, lze tvrdit, že výkon je nedostačující. Na víceprocesorových počítačích jsou ve výstupu průměrné hodnoty a pro sledování hodnot jednotlivých procesorů je třeba použít příkaz *mpstat*.^[20]

Příkaz *netstat* a jeho přepínače slouží ke sběru informací o síti. Je možné zjistit stav síťové karty, množství přijatých či odeslaných paketů, chyby na výstupu či vstupu

a kolize. Míra kolizí na síti je podílem počtu kolizí a paketů na výstupu. Jestliže je vyšší než 10% je buď přetížená či špatně nakonfigurovaná síť nebo závada na hardware.^[3]

6.1.2 Dtrace

Integrovaný dynamický nástroj DTrace poskytuje rámec pro sledování chodu jádra a uživatelských programů v produkčním prostředí. Umožňuje sledovat problémy s výkonem aplikací a detekovat případné odchylky v chování ať již interpretovaných jazyků, nativních programů, knihoven, systémových volání, tak i samotného jádra.

Tato vlastnost je implementována sondami, což jsou místa rozeseta po celém systému. DTrace je schopen na tyto sondy navázat žádosti o vykonání speciálních činností. Sondy potom za běhu poskytují data pro interpretaci nástrojem DTrace. Výpis standardních sond lze získat pomocí příkazu `dtrace -l`. Alfanumerický identifikátor sondy, který je uveden ve čtvrtém sloupci výstupu poskytuje kromě jména sondy i informace o umístění v modulu DTrace, modulu jádra či uživatelské knihovny a funkci.

Pomocí vysokoúrovňového jazyka D vyvinutého v Sun Microsystems, který je velmi podobný jazyku C++, lze vytvářet vlastní skripty pro využití sond.^[19]

6.2 Správa zdrojů

Od dnešních síťových aplikací jsou vyžadovány pružné odezvy bez ohledu na to, kolik mají spuštěných procesů, tj. jaký mají workload. Pokud není využívána správa zdrojů, operační systém standardně poskytuje zdroje stejnou měrou všem procesům. Mezi více procesními skupinami jednotlivých aplikací může dojít ke kolizi žádostí o zdroje a omezení běhu některé aplikace.

Správa zdrojů by měla zajistit jejich účinnější využití a taktéž požadované odezvy aplikací. Měla by být dosažena minimalizace kolizí žádostí o zdroje pro jednotlivé aplikace.

Vlastnosti správy zdrojů operačního systému Solaris 10 dovolují různé metody. Lze omezit přístup ke zdroji, zajistit jeho výlučnou alokaci pro jednotlivé aplikace nebo od sebe skupiny procesů jednotlivých aplikací oddělit. V operačním systému je správa zdrojů implementována sadou algoritmů ošetřujících žádosti o zdroje zasílané běžícími

aplikacemi. Nástroje správy zdrojů umožňují přizpůsobení standardního chování operačního systému při různých zatíženích běžícími aplikacemi. To může být docíleno zakázáním zdroje nebo zajištěním přednostní alokace pro určitou aplikaci. Je rovněž možno aplikovat alokační pravidla souhrnně. V systému využívajícím prostředky správy zdrojů může být ošetřena nadměrná spotřeba zdrojů jedinou aplikací, změněna priorita aplikace v závislosti na vnějších okolnostech a zdroje alokovány s ohledem na jejich efektivní využití.

Při plánování použití správy zdrojů je třeba identifikovat aplikace, jejichž procesům hrozí kolize. Rovněž je nutné určit aplikace, jejichž workload nemá efekt na preferované aplikace. Pakliže jsou identifikovány kolizní a spolupracující workloady, je možné vytvořit konfiguraci s minimálními omezeními běhu jednotlivých aplikací.

Pro účinnou správu zdrojů v operačním systému slouží kontrolní, notifikační a monitorovací mechanismy. Mnohé tyto funkce existují jako nadstavba v systému již existujících mechanismů, jiné jsou specifické.

Pomocí kontrolních mechanismů je možno efektivní správu zdrojů docílit buď omezováním zdrojů, plánováním využití zdrojů nebo jejich kompletním oddělením. Omezováním je možné nastavit hranice spotřeby zdrojů. Může ovšem dojít k takovému nastavení restrikcí, že aplikace nebude schopna běhu. Tomu lze předejít postupným omezováním zdrojů a sledováním chování aplikace. Plánování zahrnuje vytváření sledu alokačních rozhodnutí. Ta jsou prováděna na základě rozodovacího algoritmu. Aplikace, které v určitém intervalu zdroj nebudou potřebovat, z něj uvolní svoje alokace. Plánování umožňuje vyšší využití zdrojů. Posledním typem kontrolního mechanismu je izolace. Skupině procesů aplikace jsou vyhrazeny zdroje do tzv. *Resource Pools*. Při použití mechanismu izolace ovšem nedochází k plnému využití zdrojů.

Nástroje správy zdrojů mohou být rovněž konfigurovány prostřednictvím jmenných služeb, kdy omezení mohou být konfigurována na více počítačích najednou.

Využití nástrojů správy zdrojů je vhodné zejména pro aplikace konsolidované na jednom vysoce výkonném serveru. Nákladnost a obtížnost administrace více fyzických serverů vede k využívání škálovatelného hardware, kde je možné provést izolaci procesních skupin jednotlivých aplikací. Správa zdrojů tedy vede ke snižování

celkové ceny provozu technických prostředků, neboť několik aplikací, které mají různé zdrojové požadavky, může běžet na jednom fyzickém serveru.^[25]

7 Oblasti využití OS Solaris

V oblasti serverových řešení slouží Solaris 10 především jako výkonná, stabilní a bezpečná platforma pro běh síťových aplikací a databází. V kombinaci se software Sun Java Enterprise System dovoluje zabezpečit celou infrastrukturu informačních systémů. Virtualizační technologie jako Solaris Containers nebo nejnovější LDomS pro servery Sun Fire T1000 a T2000 vybavené procesory Cool Threads dovolují využít škálovatelný hardware pro bezkolizní běh více aplikací v rámci jednoho fyzického serveru.

Uvolnění zdrojového kódu Solaris 10 pod Common Development and Distribution Licence je příčinou jeho rozšíření mezi individuální uživatele a vývojáře. Na základě tohoto kroku vznikl i projekt OpenSolaris. Iniciativa je sponzorována společností Sun Microsystems, čímž dochází k urychlení vývoje nových funkcionalit operačního systému.

V rámci České republiky je operační systém Solaris implementován např. ve výrobních podnicích, finančních ústavech, vzdělávacích institucích nebo organizacích státní správy. Autorka práce se aktivně podílí na správě části technické infrastruktury nejmenovaného strojírenského podniku s automatizovanými výrobními linkami využívajícími aplikace a databáze nasazené právě na platformě Solaris 10. Pro zajištění třísměnného provozu organizace vyžaduje vysokou dostupnost nasazených aplikací a databází. Ta je zabezpečena prostřednictvím software Veritas Cluster Server 5.0. Pro správu datových úložišť pro databáze Oracle 9i a 10g je určen Veritas Volume Manager 5.0. Zatímco produkty firmy Symantec, dříve Veritas, vyžadují při snaze o zajištění běžného provozu častou interakci s výrobcem, u operačního systému Solaris 10 je tato aktivita nutná spíš při nasazování nové vlastnosti. Je zde patrný rozdíl mezi podporou operačního systému s uvolněným zdrojovým kódem a proprietárního software.

Jiným příkladem využití operačního systému je souborový server. Sdílený diskový prostor slouží jako úložiště pro podnikovou vědomostní databázi společnosti

působící v oblasti informačních technologií. Požadavkem byla přístupnost serveru pro klienty s operačními systémy Windows a Linux a možnost připojení vzdálených souborových systémů jako lokálních jednotek. Pro tyto účely byla využita skutečnost, že výše uvedené operační systémy, stejně tak jako Solaris 10, podporují protokoly SMB a CIFS. Souborový systém je využíván s poměrně vysokou frekvencí a slouží svému účelu přes dva roky. Jediným servisním zásahem bylo rozšíření jeho kapacity.

8 Závěr

Tato bakalářská práce se pokusila zmapovat téma administrace operačního systému Solaris 10. Jejím cílem bylo upevnit si základní administrátorské principy a představit specifické vlastnosti, díky kterým tento operační systém může být označen jako ideální volba pro podnikové prostředí.

Protože se jedná o velmi široké téma, byly vybrány příklady, které nejvýstižněji odlišují operační systém společnosti Sun Microsystems a hardwarovou platformu SPARC od ostatních produktů. Proto práce podává přehled o instalačních metodách, balíčkovacím systému a systému patchů. Dále byly nastíněny metody startu a zastavení operačního systému, které umožňují pružně reagovat na okamžité potřeby prostředí, kde jsou mnohdy nasazené kritické aplikace. V této souvislosti byl popsán i model Service Management Facility usnadňující administraci a diagnostiku problémů systémových služeb.

V kapitole věnované souborovým systémům, fyzickým a logickým diskům byl představen Zetta-File System, kombinace souborového systému a správce svazků, v jejímž konceptu je kladen důraz na integritu dat. Kapitola obsahuje i přehled nativních UNIXových nástrojů pro zálohování dat.

Uvedený návrh skriptu pro dávkové zpracování rutinních administrátorských činností je příkladem využití příkazového interpretu v každodenním provozu. Přestože daleko vhodnější jsou pro správu informací o uživatelských jmenné služby, jejich použití je vázáno na data statického charakteru, která nejsou modifikována s příliš vysokou frekvencí. Vytvořený skript by mohl tedy posloužit svému účelu např. ve vzdělávacích institucích pro vytváření uživatelských účtů frekventantů krátkodobých kurzů, kde se ke školícím účelům používá i výpočetní technika.

Obsahem kapitoly o výkonu operačního systému je kromě popisu nejpoužívanějších nástrojů pro monitoring výkonu systému i pojednání o dynamickém nástroji DTrace poskytujícím rámec pro sledování chodu jádra a uživatelských programů v produkčním prostředí. Umožňuje sledovat problémy s výkonem aplikací a detekovat případné odchylky v chování ať již interpretovaných jazyků, nativních programů, knihoven, systémových volání, tak i samotného jádra.

Veškeré praktické ukázky práce v příkazovém řádku a screenshot grafického prostředí byl pořízen na hardware firmy Sun Microsystems. Jednalo se o dvouprocesorový server Sun Enterprise 220R a plně 64-bitovou pracovní stanici Sun Blade 150. Oba tyto systémy jsou založené na procesorovém chipu s architekturou UltraSPARC.

Díky výše uvedeným vlastnostem je operační systém Solaris v mnoha podnikových prostředích osvědčenou stabilní a spolehlivou platformou pro běh aplikací a databází, jak ostatně dokládají i uvedené příklady z praxe. Nicméně jeho správa vyžaduje mnoho času, zejména ve fázi po instalaci pro získání potřebných parametrů. Investovaný čas se ale zúročí ve výkonu a spolehlivosti. Napomůže tomu i zevrubná dokumentace na webových stránkách firmy Sun Microsystems.

9 Seznamy

9.1 Literatura a ostatní zdroje

- [1] BACH, Maurice J. The Design Of The Unix Operating System. New Jersey: Prentice-Hall PTR 1986. 486 p. ISBN 0-13-201799-7
- [2] CALKINS, William. Excerpt from the Solaris 8 System Administrator Certification Training Guide [online]. [cit:2009-03-23]. Dostupné z URL <http://unixed.com/Resources/history_of_solaris.pdf>
- [3] CALKINS, William. Solaris 10 System Administration Exam Prep. 1st ed. Exam Cram 2005. ISBN: 0-7897-3461-3
- [4] GÁLA, Libor & spol. Podniková informatika, Praha: Grada Publishing 2005. ISBN 80-247-1278-4
- [5] GOLDMAN, Gary. UltraSPARC™: The Advancement of UltraComputing [online] Sun Microsystems. [cit: 2009-03-24]. Dostupné z URL <<http://infopad.eecs.berkeley.edu/CIC/archive/UltraSparc2.CompCon96.pdf>>
- [6] KAPOUN, Jan. Pohledy do historie světových IT firem (8): Sun Microsystems [seriál online]. Praha: IDG COMMUNICATIONS, 2008-08-22. [cit: 2009-03-23]. Dostupné z URL <<http://scienceworld.cz/ekonomika/pohledy-ho-historie-svetovych-it-firem-8-sun-microsystems-401>>
- [7] KURFIRST, Michal. Historie operačních systémů Windows, Unix, Mac OS a Linux. MujMAC [online]. 2006-04-13. [cit: 2009-03-15]. Dostupné z URL <<http://www.muymac.cz/art/polemiky/historie-operacnich-systemu-win-unix-macosx.html>>
- [8] MORGAN, Timothy Prickett. Solaris 10 Is All About Performance [online]. THE UNIX GUARDIAN. 2004-12-02. [cit: 04-04-2009]. Volume 1. Number 43. Dostupné z URL <<http://www.itjungle.com/tug/tug120204-story01.html>>
- [9] RITCHIE, Dennis. Yes, A video game contributed to Unix Development [online]. [cit: 2009-03-15]. Dostupné z URL <<http://www.fas.harvard.edu/~lib215/reference/history/spacetravel.html>>

- [10] SCHUWEILER, Greg. The „format“ Utility in the Solaris Operating System [online]. Sun Microsystems 2004-11-10. [cit: 2009-04-04]. Dostupné z URL <http://www.sun.com/bigadmin/content/submitted/format_utility.jsp>
- [11] SKOČOVSKÝ, Luděk, Principy a problémy operačního systému UNIX. 2. vydání. Brno: Luděk Skočovský 2008. 288 s. ISBN 80-902612-5-6
- [12] VESELÝ, Arnošt, OPERAČNÍ SYSTÉMY II. 1. vydání. Praha: CREDIT 2001, 258 s. ISBN 80-213-0773-0
- [13] WATTERS, Paul A. Solaris 10: The Complete Reference. 1st ed. McGraw-Hill Companies 2005. 738 p. ISBN-10:0-07-146657-6
- [14] BSD. Wikipedie [online].2009-02-23.[cit. 2009-03-14]. Dostupné z URL <<http://cs.wikipedia.org/wiki/BSD>>
- [15] Datasheet Sun Blade 150 Workstation [online] Sun Microsystems 2003. [cit: 2009-03-24] Dostupné z URL <<http://www.sun.com/sunblade150>>
- [16] man pages section 1: User Commands [online]. Sun Microsystems 2002-05-21. [cit: 2009-03-31]. Dostupné z URL <<http://docs.sun.com/app/docs/doc/816-0210/>>
- [17] man pages section 1M: System Administration Commands [online]. Sun Microsystems 2008-04-01. [cit: 2009-03-31]. Dostupné z URL <<http://docs.sun.com/app/docs/doc/817-3937/>>
- [18] Overview of UltraSPARC Processors [online]. [cit: 2009-03-23]. Dostupné z URL <<http://www.sun.com/processors/UltraSPARC-T1/>>
- [19] Solaris 10 What`s New [online]. Sun Microsystems 2008. [cit: 2009-04-05]. Dostupné z URL <<http://dlc.sun.com/pdf/817-0547/817-0547.pdf>>
- [20] Solaris Performance Monitoring & Tuning – iostat, vmstat & netstat. [online]. Admin`s Choice 2007-01-21. [cit: 2009-04-05]. Dostupné z URL <http://www.adminschoice.com/docs/iostat_vmstat_netstat.htm>
- [21] Solaris Volume Manager Administration Guide [online]. Sun Microsystems 2006. [cit: 2009-04-04]. Dostupné z URL <<http://dlc.sun.com/pdf/816-4520/816-4520.pdf>>
- [22] Solaris ZFS Administration Guide [online]. Sun Microsystems 2008. [cit: 2009-04-04]. Dostupné z URL <<http://dlc.sun.com/pdf/819-5461/819-5461.pdf>>

- [23] Sun Enterprise 220R server: Hardware Specifications [online]. Sun Microsystems 2008. [cit: 2009-03-24]. Dostupné z URL <http://sunsolve.sun.com/handbook_pub/validateUser.do?target=Systems/E220R/spec>
- [24] System Administration Guide: Naming and Directory Services (DNS, NIS and LDAP) [online]. Sun Microsystems 2008. [cit: 2009-04-04]. Dostupné z URL <<http://dlc.sun.com/pdf/816.4556.pdf>>
- [25] System Administration Guide: Virtualization Using the Solaris Operating System [online]. Sun Microsystems 2008. [cit: 2009-04-06]. Dostupné z URL <<http://dlc.sun.com/pdf/819-2450/819-2450.pdf>>
- [26] System Requirements [online] Sun Microsystems 2007-09-05. [cit: 2009-03-31]. Dostupné z URL <<http://www.sun.com/software/solaris/specs.jsp>>
- [27] UNIX System V.Wikipedia [online].2009-03-04.[cit. 2009-03-14]. Dostupné z URL http://en.wikipedia.org/wiki/UNIX_System_V
- [28] What are SunOS and Solaris? [online]. Indiana Univeristy 2005-11-25. [cit: 2009-03-23]. Dostupné z URL <<http://kb.iu.edu/data/agjq.html>>

9.2 Obrázky

Obrázek 1: Úvodní obrazovka Solaris Management Console verze 2.1..... 19

9.3 Tabulky

Tabulka 1: Zkrácený výpis systémové konfigurace serveru macek.....	17
Tabulka 2: Zkrácený výpis systémové konfigurace pracovní stanice sunray	18
Tabulka 3: Použití příkazu pkginfo s přepínačem -l pro dlouhý výpis informací o softwarovém balíčku	21
Tabulka 4: Výstup příkazu "showrev -p"	22
Tabulka 5: Pracovní stanice sunray má povoleno automatické zavedení operačního systému, server macek nikoliv	23
Tabulka 6: Příklad služeb řízených init skripty; zakázaných a běžících v rámci SMF...	25
Tabulka 7: Výpis běžících služebních skupin.....	25
Tabulka 8: Výpis aktuálního provozního režimu.....	26

Tabulka 9: Nabídka interaktivního příkazu "format"	27
Tabulka 10: Nabídka úkolů, které lze nad diskem provést pomocí interaktivního příkazu "format"	28
Tabulka 11: Tabulka /etc/vfstab popisující vlastnosti souborových systémů	29
Tabulka 12: Informace o uživatelských účtech v /etc/passwd	32
Tabulka 13: Vytvoření nového uživatelského účtu	32
Tabulka 14: Založení nového uživatele a záznamy v příslušných souborech	32
Tabulka 15: Zobrazení všech dostupných síťových rozhraní v systému	37

10 Přílohy

Příloha č. 1 - *skript příkazového interpretu bash pro dávkové vytvoření uživatelských účtů ze seznamu uživatelů*

```
oot@sunray / # cat /skripty/newuser.sh
# !/bin/bash
### definice promennych #####
# USER_LIST : seznam uzivatelu
USER_LIST=/export/home/vjanovsk/prvaci
# USER_LIST_H : pomocny seznam uzivatelu
USER_LIST_H=$USER_LIST"_H"
# n : poradí v seznamu studentu
# i : slouží pro iteraci v generatoru loginu
# first_last_vol : radek ze souboru USER_LIST
# last_vol : radek zkraceny o prvni atribut
# FIRST : krestni jmeno
# LAST : prijmeni
# VOL : rocnik narozeni
# F : iniciala z $FIRST
# L : prvni tri litery z $LAST
# V :
# DATE : rok zapoceti studia
# UNCD : kod zarucujici unikatnost loginu $n + $DATE
# LOGIN : login uzivatele
### konec definice promennych #####

# polozky v seznamu uzivatele s atributy 'jmeno', 'prijmeni'
# 'rocnik_narozeni' oddeleny carčkama (" , ")
# predpoklad spravneho souboru uzivatele, tj. zadne znaky navic

### kontrola existence seznamu uzivatele $USER_LIST
if [ ! -f $USER_LIST ]; then
    echo "Seznam uzivatele $USER_LIST neexistuje"
    exit 1
fi

### kontrola existence pomocneho seznamu $USER_LIST_H
if [ -f $USER_LIST_H ]; then
    cat /dev/null > $USER_LIST_H
fi

### konec kontroly existence pomocneho seznamu
### generator loginu #####
n=0
DATE=`date +%y`
for i in `cat $USER_LIST`
do
    first_last_vol=$i
    last_vol=${first_last_vol#*,}
    FIRST=${first_last_vol%*\,}
    F=`echo $FIRST|cut -c 1|tr -s '[:upper:]' '[:lower:]'`
    LAST=${last_vol%*,}
    L=`echo $LAST|cut -c 1-3|tr -s '[:upper:]' '[:lower:]'`

```

```
VOL=${last_vol##*,}
n=`expr $n + 1`
UNCD=`expr $DATE + $n`
LOGIN=$L,$F$UNCD$VOL
echo "$LAST,$FIRST,$LOGIN" >> $USER_LIST_H
useradd -s /bin/bash -m -d /export/home/$LOGIN -g 10 -c
"$FIRST $LAST" $LOGIN
passwd -df $LOGIN
done
```