

Módulo 8

Sistema Operacional



Lição 8

Regiões no Solaris

Versão 1.0 - Mar/2008

Autor

-

Equipe

Rommel Faria

John Paul Petines

Necessidades para os Exercícios**Sistemas Operacionais Suportados****NetBeans IDE 5.5** para os seguintes sistemas operacionais:

- Microsoft Windows XP Professional SP2 ou superior
- Mac OS X 10.4.5 ou superior
- Red Hat Fedora Core 3
- Solaris™ 10 Operating System (SPARC® e x86/x64 Platform Edition)

NetBeans Enterprise Pack, poderá ser executado nas seguintes plataformas:

- Microsoft Windows 2000 Professional SP4
- Solaris™ 8 OS (SPARC e x86/x64 Platform Edition) e Solaris 9 OS (SPARC e x86/x64 Platform Edition)
- Várias outras distribuições Linux

Configuração Mínima de Hardware**Nota:** IDE NetBeans com resolução de tela em 1024x768 pixel

Sistema Operacional	Processador	Memória	HD Livre
Microsoft Windows	500 MHz Intel Pentium III workstation ou equivalente	512 MB	850 MB
Linux	500 MHz Intel Pentium III workstation ou equivalente	512 MB	450 MB
Solaris OS (SPARC)	UltraSPARC II 450 MHz	512 MB	450 MB
Solaris OS (x86/x64 Platform Edition)	AMD Opteron 100 Série 1.8 GHz	512 MB	450 MB
Mac OS X	PowerPC G4	512 MB	450 MB

Configuração Recomendada de Hardware

Sistema Operacional	Processador	Memória	HD Livre
Microsoft Windows	1.4 GHz Intel Pentium III workstation ou equivalente	1 GB	1 GB
Linux	1.4 GHz Intel Pentium III workstation ou equivalente	1 GB	850 MB
Solaris OS (SPARC)	UltraSPARC IIIi 1 GHz	1 GB	850 MB
Solaris OS (x86/x64 Platform Edition)	AMD Opteron 100 Series 1.8 GHz	1 GB	850 MB
Mac OS X	PowerPC G5	1 GB	850 MB

Requerimentos de Software

NetBeans Enterprise Pack 5.5 executando sobre Java 2 Platform Standard Edition Development Kit 5.0 ou superior (JDK 5.0, versão 1.5.0_01 ou superior), contemplando a Java Runtime Environment, ferramentas de desenvolvimento para compilar, depurar, e executar aplicações escritas em linguagem Java. Sun Java System Application Server Platform Edition 9.

- Para **Solaris, Windows, e Linux**, os arquivos da JDK podem ser obtidos para sua plataforma em <http://java.sun.com/j2se/1.5.0/download.html>
- Para **Mac OS X**, Java 2 Platform Standard Edition (J2SE) 5.0 Release 4, pode ser obtida diretamente da Apple's Developer Connection, no endereço: <http://developer.apple.com/java> (é necessário registrar o download da JDK).

Para mais informações: <http://www.netbeans.org/community/releases/55/relnotes.html>

Colaboradores que auxiliaram no processo de tradução e revisão

Aécio Júnior	Carlos Fernandes Gonçalves	Massimiliano Girolodi
Alberto Ivo da Costa Vieira	Denis Mitsuo Nakasaki	Paulo Oliveira Sampaio Reis
Alexandre Mori	Felipe Gaúcho	Ronie Dotzlaw
Alexis da Rocha Silva	Jacqueline Susann Barbosa	Seire Pareja
Allan Wojcik da Silva	João Vianney Barrozo Costa	Thiago Magela Rodrigues Dias
Antonio José Rodrigues Alves Ramos	Luiz Fernandes de Oliveira Junior	Vinícius Gadis Ribeiro
Angelo de Oliveira	Marco Aurélio Martins Bessa	
Bruno da Silva Bonfim	Maria Carolina Ferreira da Silva	

Auxiliadores especiais

Revisão Geral do texto para os seguintes Países:

- **Brasil** – Tiago Flach
- **Guiné Bissau** – Alfredo Cá, Bunene Sisse e Buon Olossato Quebi – ONG Asas de Socorro

Coordenação do DFJUG

- **Daniel deOliveira** – JUGLeader responsável pelos acordos de parcerias
- **Luci Campos** - Idealizadora do DFJUG responsável pelo apoio social
- **Fernando Anselmo** - Coordenador responsável pelo processo de tradução e revisão, disponibilização dos materiais e inserção de novos módulos
- **Rodrigo Nunes** - Coordenador responsável pela parte multimídia
- **Sérgio Gomes Veloso** - Coordenador responsável pelo ambiente JEDI™ (Moodle)

Agradecimento Especial

John Paul Petines – Criador da Iniciativa JEDI™

Rommel Feria – Criador da Iniciativa JEDI™

Original desta por – McDougall e Mauro – Solaris Internals. Sun Microsystems. 2007.

1. Objetivos

Uma região (*zone*) é um ambiente virtualizado no sistema operacional criado por uma única instância. Na essência, é criado um "computador dentro de um computador"; um sistema completamente virtual que existe sobre o sistema físico.

Nesta seção, discutiremos as regiões do Solaris. Começaremos com uma discussão sobre o que às regiões são e quais vantagens de sua utilização sobre um sistema provedor de serviços. Essa lição fornece uma visão geral ao mostrar quais comandos são necessários para criar uma região. Finalmente, terminaremos com uma discussão sobre algumas opções de regiões comumente utilizadas.

Ao final desta lição, o estudante será capaz de:

- Explicar o que são regiões no Solaris
- Criar uma região básica
- Estar familiarizado com alguns comandos avançados para a criação de regiões

2. Introdução

2.1. Utilização típica para regiões

As regiões permitem múltiplas instâncias separadas do Solaris existindo independentemente uma da outra em um só computador, o que pode ser bem vantajoso em certas situações. Considere um escritório que necessite de um servidor WEB, de banco de dados, de desenvolvimento, de email e de arquivo. Essa companhia possui duas opções:

- **Opção 1:** Um computador por serviço. Uma companhia executando esta opção não teria escolha a não ser obter cinco computadores.
- **Opção 2:** Ter todos esses serviços em um só computador ou mais de um serviço por computador. Nessa opção, há margem para haver alguns problemas de configuração, particularmente se alguns desses serviços tiverem conflitos uns com os outros.

As regiões permitem uma terceira opção. Os serviços podem ser instalados cada um em sua própria instância do Solaris, sendo executados em um único computador físico. Isso combina o melhor dos dois mundos. Primeiramente, não há mais necessidade de comprar computadores múltiplos já que as regiões permitem computadores virtuais sendo executados em um só sistema autônomo. Em segundo lugar, como cada serviço existe em seu próprio sistema autônomo, não há mais necessidade de se preocupar com conflitos de configuração; um serviço pode ser instalado em uma região que está configurada precisamente com as necessidades desse serviço.

2.2. Vantagens

A utilização de regiões fornece as seguintes vantagens:

2.2.1. Segurança

Se um serviço de rede é feito para ser executado em uma região, invasores que pretendam utilizar as vulnerabilidades deste serviço para ganhar acesso ao sistema somente poderão acessar a região do serviço. Qualquer dano causado será limitado a essa região. Acessar outra região seria como invadir um segundo computador, apesar do fato de que essas regiões na verdade são executadas no mesmo computador físico.

Uma região pode ter usuários diferentes de outras regiões. O administrador do sistema global pode escolher dar acesso aos usuários apenas a serviços específicos e não a todo sistema.

A utilização de regiões permite um segundo nível de administração, o superusuário da região. Este usuário tem direitos de administrar apenas para uma região específica. Isso permite que o administrador do sistema global (o superusuário da região global principal) delegue tarefas administrativas da região de outros usuários enquanto mantém a habilidade de promover as grandes mudanças necessárias no sistema.

2.2.2. Isolamento

As regiões trabalham isoladamente, não estão cientes da existência de outras regiões. Desta forma, processos sendo executados em uma determinada região são incapazes de influenciar processos em outras regiões. A configuração de uma região é importante apenas para esta região e não afeta outras regiões. Uma aplicação sendo executada em uma região pode ter o benefício de ser executada em um ambiente criado especificamente para ela, por isso, não há mais necessidade de preocupações com detalhes de configurações.

2.2.3. Virtualização

A configuração do sistema global é irrelevante para uma região. As regiões podem ter suas próprias customizações de ambiente virtual. Este ambiente pode até mesmo ser copiado em diferentes máquinas, permitindo assim que uma região tenha o mesmo ambiente virtual mesmo em diferentes máquinas com diferentes configurações.

2.2.4. Granularidade

Há muitas opções disponíveis para configurar o nível de isolamento de uma região. Uma região pode ter sua própria CPU, memória, espaço de disco rígido, ou ter estes recursos compartilhados entre diferentes regiões mesmo sem saber que nenhum compartilhamento está sendo feito.

2.2.5. Ambiente

Uma aplicação não precisa ser recompilada para ser executada em outra região. O Solaris fornece as mesmas bibliotecas e interfaces para todas as regiões. Existem apenas algumas restrições que não permitem programas executar grandes mudanças enquanto dentro de uma região.

Adicionalmente, uma aplicação não é afetada pela existência de outras regiões, e é praticamente inconsciente dessas outras regiões. As regiões são criadas para que uma aplicação possa ser executada em uma instalação autônoma do Solaris.

2.3. Os dois tipos de região

Existem dois tipos de região no Solaris. A região **global** e as regiões **não-globais**.

A **região global** é a instalação padrão que é utilizada pelo Solaris. Toda a administração do sistema é executada a partir desta região. Esta é a única região que está ciente de todas as outras regiões que estão sendo executadas no sistema. É a partir desta região que a administração de regiões é realizada.

As **regiões não-globais** são as regiões adicionais que são configuradas a partir da região global. Essas são as regiões que são executadas sobre a região global. As regiões não-globais não estão cientes de que são regiões ou da existência de outras regiões. Assim, a manutenção desta região não é possível a partir de uma região não-global.

O **superusuário** em uma região global pode ser considerado o superusuário global, habilitado a executar grandes mudanças no sistema, incluindo mudanças que afetam as outras regiões. O superusuário em uma região não-global só pode administrar aquela região.

2.4. Mapeamento de tarefas por região

Para configurar regiões para um computador, as seguintes tarefas precisam ser realizadas:

1. Identificar quais aplicações serão executadas na região.
2. Determinar quantas regiões serão configuradas (pode-se ter um máximo de 8.192 regiões. Quanto menos regiões, melhor para o sistema).
3. Determinar se uma região requer características de gerenciamento avançado de recursos.
4. Executar os passos de pré-configuração, tais como, determinar o endereço de IP da região, nome do servidor, usuários para esta região, entre outros.
5. Gravar a configuração da região utilizando o comando **zonecfg**.
6. Configurar e instalar a região.
7. Carregar a região utilizando o comando **zoneadm**.
8. Acessar a região utilizando **zlogin** e customizar cada uma de acordo com o planejamento inicial de configuração.

3. Configurando uma região básica

Veremos agora como configurar uma região básica no Solaris. Mostraremos os comandos necessários para realizar esta tarefa e conheceremos mais informações sobre regiões enquanto a configuramos.

3.1. Iniciação do **zonecfg**

Para configurar uma região utilizaremos o comando **zonecfg**. Para executar este comando é necessário estar conectado como superusuário e executá-lo com as seguintes opções:

```
# zonecfg -z myfirstzone
```

A opção **-z** indica o nome da região que se deseja configurar; neste caso, **myfirstzone**. Após executar este comando, a tela exibirá o alerta do **zonecfg** do seguinte modo:

```
zonecfg:myfirstzone>
```

Indicando que o **zonecfg** está sendo executado e pronto para receber comandos. Configurar uma região quer dizer executar comandos **zonecfg**. Como o alerta indica, esses comandos serão utilizados para configurar **myfirstzone**.

Há dois modos do **zonecfg**:

1. **Modo global** – comandos **zonecfg** que são executados no modo global são comandos que configuram os atributos da região, tais como onde a região deve ser instalada ou se uma região deve ou não ser automaticamente carregada ao iniciar.
2. **Modo recurso** – significa que os próximos comandos são utilizados para descrever recursos na região, tais como os arquivos de sistema ou rede.

Configurar uma região implica alternar entre estes dois modos, executando comandos que indicam os atributos da região e comandos que descrevem seus recursos. Quando o **zonecfg** é iniciado, está automaticamente no modo global.

3.2. Configuração de região básica

A figura seguinte mostra os diferentes comandos necessários para configurar **myfirstzone**. A seguir discutiremos estes comandos de forma mais detalhada.

```
zonecfg:myfirstzone> create
zonecfg:myfirstzone> set zonepath=/export/home/myfirstzone
zonecfg:myfirstzone> set autoboot=true
zonecfg:myfirstzone> add fs
zonecfg:myfirstzone:fs> set dir=/usr/local
zonecfg:myfirstzone:fs> set special=/opt/local
zonecfg:myfirstzone:fs> set type=lofs
zonecfg:myfirstzone:fs> end
zonecfg:myfirstzone> add net
zonecfg:myfirstzone:net> set address=192.168.0.1/24
zonecfg:myfirstzone:net> set physical=hme0
zonecfg:myfirstzone:net> end
zonecfg:myfirstzone> add attr
zonecfg:myfirstzone:attr> set type=string
zonecfg:myfirstzone:attr> set value="my first zone"
zonecfg:myfirstzone:attr> end
zonecfg:myfirstzone> verify
zonecfg:myfirstzone> commit
zonecfg:myfirstzone> exit
```

3.3. Criação da configuração de uma região

Inicia-se a criação da configuração de um região com o comando **create**.

```
zonecfg:myfirstzone> create
```

Caso seja informado o nome de uma região não existente no comando **zonecfg**, o sistema solicitará a execução do comando **create**.

3.4. Configuração da raiz de diretório da região

A raiz de diretório de uma região existe em um subdiretório do arquivo de sistema da região global. Podemos especificá-la através do comando **zonepath**:

```
zonecfg:myfirstzone> set zonepath=/export/home/myfirstzone
```

O comando informa ao **zonecfg** que a raiz do diretório **myfirstzone** estará localizada em /export/home/myfirstzone. Necessariamente este diretório deve existir.

3.5. Carregamento automático

O comando **set autoboot** especifica quando uma região será ou não iniciada automaticamente quando a região global for iniciada.

```
zonecfg:myfirstzone> set autoboot=true
```

Se esta opção não for enviada, então a região deverá ter uma inicialização manual usando o comando **zoneadm**.

3.6. Arquivo de sistema da região

Por padrão, certos diretórios da região global são herdados pelas regiões não-globais para fornecer um sistema de trabalho à região. O diretórios **inherit-pkg-dir** (/lib, /platform, /sbin e /usr) não são copiados para dentro da região. O que a região recebe é um endereço somente de leitura para esses diretórios na região global. Permite que a região economize espaço de arquivo. Fora isso, quaisquer mudanças na região global particularmente para estes diretórios também refletirão nos arquivos de sistema das regiões não-globais.

Para adicionar arquivos de sistemas, utilizamos o comando **add fs**:

```
zonecfg:myfirstzone> add fs
```

O comando **add fs** alterna do modo global para o modo de recurso. Como dito, o modo de recurso significa que comandos adicionais especificados são utilizados para descrever um recurso para a região. O alerta **zonecfg** também muda para mostrar que está sendo editado um recurso de arquivo de sistema.

```
zonecfg:myfirstzone> add fs
zonecfg:myfirstzone:fs>
```

3.7. Adicionar um sistema de arquivos

A seguir estão os comandos para definir um recurso de sistema de arquivos para **myfirstzone**:

```
zonecfg:myfirstzone> add fs
zonecfg:myfirstzone:fs> set dir=/usr/local
zonecfg:myfirstzone:fs> set special=/opt/local
zonecfg:myfirstzone:fs> set type=lofs
zonecfg:myfirstzone:fs> end
```

O comando especial **set** especifica qual diretório da região global deve ser adicionado. Este diretório é colocado na região de sistema de arquivos no diretório especificado pelo **set dir**. No nosso exemplo, /opt/local, situado na região global, é colocado em /usr/local em **myfirstzone**.

O comando **set type** indica qual o tipo de sistema de arquivos do diretório a ser montado. O tipo de sistema de arquivos explica como o núcleo lidará com este diretório. O sistema de arquivos virtual loopback (**lofs**) indica que o diretório montado é o mesmo diretório no sistema de arquivos

global, entretanto, é acessado através de um caminho diferente. Isto é semelhante ao **inherit-pkg-directories**.

Devido a isso, todas as mudanças feitas no diretório global refletem também no diretório da região. Para o exemplo, se um arquivo for adicionado ao diretório **/opt/local** na região global seguido de **/usr/local** em **myfirstzone** este irá mudar também.

Observamos que o diretório **/opt/local** na região global deve ser um diretório existente.

Para terminar a configuração do sistema de arquivo, temos o comando:

```
zonecfg:myfirstzone:fs> end
zonecfg:myfirstzone>
```

Isto muda a **zonecfg** de volta ao modo global como indicado pela mudança do *prompt*. Podemos agora adicionar novos sistemas de arquivos com comando **add fs**.

3.8. Adicionar uma rede

Uma região pode ter seu próprio endereço IP individual. Não há necessidade de interfaces de rede adicionais para cada novo endereço IP definido; a região global gerencia automaticamente o roteamento. Podemos usar o comando **add net** para definir este gerenciamento.

Vejam os a sequência de instruções para configurar a rede **myfirstzone**.

```
zonecfg:myfirstzone> add net
zonecfg:myfirstzone:net> set address=192.168.0.1/24
zonecfg:myfirstzone:net> set physical=hme0
zonecfg:myfirstzone:net> end
```

Nosso exemplo configura o endereço **IP** de **myfirstzone** para **192.168.0.1** com uma máscara de rede definida como **/24** (255.255.255.0). Este endereço **IP** está vinculado a uma placa de rede **hme0** (este resultado pode ser diferente nas diversas configurações).

3.9. Descrição da região

Um recurso para a especificação da região é o comando **add attr**, que podemos utilizar para acrescentar um comentário para a região.

```
zonecfg:myfirstzone> add attr
zonecfg:myfirstzone:attr> set type=string
zonecfg:myfirstzone:attr> set value="my first zone"
zonecfg:myfirstzone:attr> end
```

3.10. Finalizar a configuração

Podemos utilizar os seguintes comandos para finalizar a configuração.

```
zonecfg:myfirstzone> verify
zonecfg:myfirstzone> commit
zonecfg:myfirstzone> exit
```

O comando **verify** verifica se a configuração da região foi digitada corretamente. Este comando permite visualizar erros. O comando **commit** salva a configuração da região. A região ainda não está instalada, apenas a configuração é armazenada. E o comando **exit** sai de **zonecfg**.

3.11. Eliminar atributos

Se errar a configuração de uma região, é possível utilizar o comando **remove**. Este comando possui duas maneiras que dependem da modalidade de **zonecfg**.

No modo global, **remove <resource-type>** especifica a eliminação de um determinado recurso. Por exemplo, o seguinte comando elimina o comentário:

```
zonecfg:myfirstzone> remove attr
```

No modo de recurso, **remove** **<property-name>** **<property-value>** elimina um elemento daquela descrição dos recursos. Por exemplo, podemos eliminar um endereço **IP** com este comando. Para fazer isso precisamos estar no recurso **net**.

```
zonecfg:myfirstzone:net> remove address=192.168.0.1/24
```

Este é o mesmo formato do comando **remove** quando se deseja eliminar qualquer um dos atributos globais. O exemplo a seguir elimina o atributo **autoboot**:

```
zonecfg:myfirstzone> remove autoboot=true
```

Quaisquer alterações feitas à configuração devem ser atualizadas.

3.12. Comandos zonecfg adicionais

Para visualizar a configuração de uma região, pode-se utilizar o comando **export**.

O comando **zonecfg** inclui um auxílio para seus comandos internos, que podemos utilizar para pedir informações adicionais. Digitando simplesmente **help** exibirá todos seus comandos. Digitando **help** **<comando>** exibirá todas as informações sobre um comando específico.

4. Zoneadm

O comando **zoneadm** é usado para administrar as regiões. Este comando vem com sub-comandos que são utilizados para indicar o que fazer com uma região específica (por meio da opção **-z**). Discutiremos esses comandos com os quais continuaremos com a configuração básica da região.

4.1. Instalação e inicialização

Depois de uma região ter sido configurada corretamente, esta pode ser instalada através do comando:

```
# zoneadm -z myfirstzone install
```

O sistema irá instalar a **myfirstzone**. Na tela serão mostrados os pacotes que estão sendo copiados para a nova região. Após o término da instalação, podemos iniciar a região através do comando:

```
# zoneadm -z myfirstzone boot
```

Se não definirmos a propriedade da região para um **autoboot**, este deverá ser o comando que utilizaremos para iniciar a região.

4.2. Listagem de regiões

Para listar as regiões existentes podemos utilizar o sub-comando **list**:

```
#zoneadm list -v
ID      NAME      STATUS
0       global      running
1       myfirstzone installed
```

O status pode assumir um dos seguintes valores para a região:

- **configured** – foi devidamente configurada com **zonecfg**, entretanto ainda não foi instalada
- **incomplete** – está no processo de instalação
- **installed** – foi instalada, entretanto ainda não está funcionando (não foi atribuída uma plataforma virtual à região)
- **ready** – foi atribuída a uma plataforma virtual e está pronta para aceitar processos de usuário
- **running** – está rodando com processos de usuário
- **shutting down** – está sendo encerrada

4.3. Comandos zoneadm adicionais

- **zoneadm -z myfirstzone halt** – interrompe a região (e a coloca em estado *ready*)
- **zoneadm -z myfirstzone reboot** – reinicializa a região
- **zoneadm -z myfirstzone uninstall** – remove a instalação da região
- **zoneadm -z myfirstzone delete** – remove a configuração da região

5. Logging in

Nesta fase, a instalação de **myfirstzone** ainda não está concluída. Para concluirmos sua instalação, devemos acessar a região através do comando:

```
# zlogin -C myfirstzone
```

A opção **-C** é a forma para acessar uma região que não está em estado *running*. Mais tarde, quando a instalação for concluída, não será mais necessário informar esta opção.

Como primeiro acesso, será necessário inserir informações sobre o sistema, semelhante às caixas de diálogo encontradas durante a instalação do Solaris. São feitas perguntas como a linguagem do sistema, o *hostname* da região, a senha do administrador (para o administrador da região). Depois desse diálogo ter sido concluído, a região vai agora entrar em estado *running* e pode aceitar processos do usuário.

Nesta fase, é possível acessar uma região e realizar sua personalização. Tais como, adicionar novos usuários, instalar novos programas e configurar os serviços que serão necessários.

6. Eliminar uma região

Executamos os seguintes passos para eliminar uma região.

Primeiro devemos encerrar a região:

```
# zlogin myfirstzone shutdown
```

Em seguida, eliminar a instalação da região. A opção **-F** corresponde forçar uma desinstalação:

```
# zoneadm -z myfirstzone uninstall -F
```

E eliminar a configuração da região:

```
# zoneadm -z myfirstzone delete -F
```

Parceiros que tornaram JEDI™ possível



Instituto CTS

Patrocinador do DFJUG.

Sun Microsystems

Fornecimento de servidor de dados para o armazenamento dos vídeo-aulas.

Java Research and Development Center da Universidade das Filipinas

Criador da Iniciativa JEDI™.

DFJUG

Detentor dos direitos do JEDI™ nos países de língua portuguesa.

Politec

Suporte e apoio financeiro e logístico a todo o processo.

Instituto Gaudium

Fornecimento da sua infra-estrutura de hardware de seus servidores para que os milhares de alunos possam acessar o material do curso simultaneamente.