Módulo 8

Sistema Operacional

Lição 4

Processo no Solaris

Versão 1.0 - Mar/2008

Autor
-

Equipe
Rommel Feria
John Paul Petines

Necessidades para os Exercícios

Sistemas Operacionais Suportados

NetBeans IDE 5.5 para os seguintes sistemas operacionais:

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

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

Um processo pode ser definido como um programa em execução. Conforme já discutido anteriormente, programas existem apenas como instruções em um arquivo até que sejam executados pela CPU. Quando um programa é executado, as instruções são carregados para a memória principal e, assim, torna-se um processo. Este capítulo discute como Solaris implementa processos.

Ao final desta lição, o estudante será capaz de:

- Conhecer sobre os componentes de um processo
- Compreender como funciona a estrutura de processos no Solaris

2. Componentes do Processo

Um serviço que está sendo executado no sistema Solaris é definido como **processo**. O sistema operacional mantém o controle de múltiplos processos. Estes podem ser provenientes de um usuário com múltiplos processos ou de múltiplos usuários de um modo simultâneo. Cada processo, no sistema Solaris, recebe uma identificação única, denominada **PID**. Uma lista de processos é armazenada em uma tabela de processos. A tabela de processo pode ser visualizada através do comando **ps** (***p****roces****s****ses*).

```
# ps -ef

 UID   PID PPID C    STIME TTY  TIME CMD
root     0    0 0   Nov 20 ?    0:11 sched
root     1    0 0   Nov 20 ?    0:02 /sbin/init
root     2    0 0   Nov 20 ?    0:00 pageout
root     3    0 0   Nov 20 ?   11:05 fsflush
root   215    1 0   Nov 20 ?    0:00 /usr/sbin/cron
root     7    1 0   Nov 20 ?    0:13 /lib/svc/bin/svc.startd
root     9    1 0   Nov 20 ?    0:28 /lib/svc/bin/svc.configd
root   124    1 0   Nov 20 ?    0:26 /usr/sbin/nscd
root   101    1 0   Nov 20 ?    0:00 /usr/lib/snmp/snmpdx -y -c /etc/snmp/conf
root 1840 1836 0 22:17:30 pts  0:00 sh
```

Um processo pode consistir de vários serviços (*threads*) de usuários. Um serviço é um trecho de código em execução que roda dentro de um processo. Um processo tradicional (tal como um programa em linguagem Java) teria um único serviço executando. Um processo em Solaris pode suportar mais de um serviço executando simultaneamente (característica *multithread*). Posteriormente discutiremos a forma de programar aplicações *multithread*.

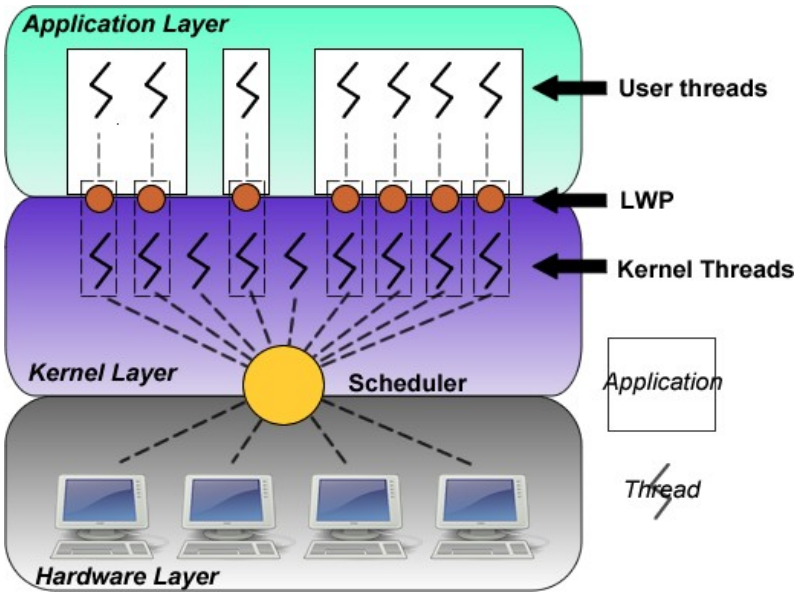

Figura 1: Processos do Sistema Solaris

Cada processo do usuário está associado no núcleo do sistema Solaris através de um *Light Weight Process* ou **LWP**. Um **LWP** permite que cada processo acesse as funções do *Kernel* de modo independente de outros processos.

Cada **LWP** é executado por um *Kernel Thread*. O *Kernel Thread* é a menor unidade de sincronização no Solaris. Em essência, processos, embora construídos com **LWP**, serão executados no *Kernel Threads*.

De modo a otimizar o tempo de início de um processo, o *Kernel* mantém **LWP's** sempre prontos para aceitar um novo processo.

3. Estrutura de Processos

O Solaris armazena informações sobre cada processo em execução. A informação armazenada é resumida pela figura a seguir.

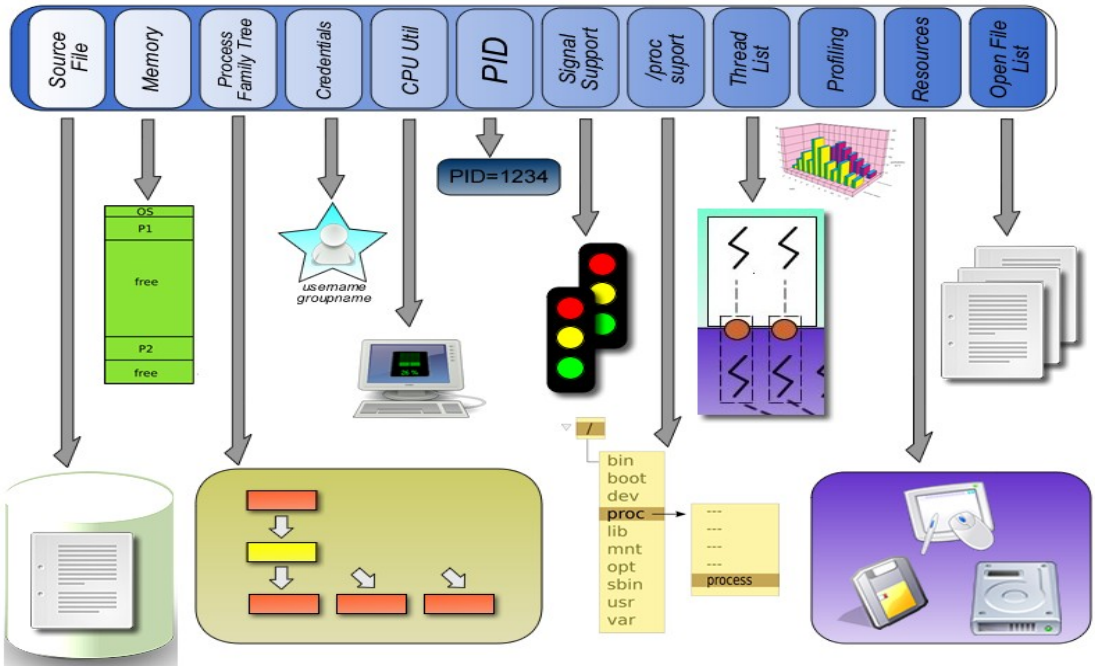

Figura 2: Sistema de Permissões

- ***Source File*** - contém as informações sobre o arquivo executável deste processo.
- ***Memory*** – indica as páginas de memória onde reside o processo.
- ***Process Family Tree*** – mostra toda a estrutura de processos, o processo que criou este (processo pai) e os processos filhos gerados por este.
- ***Credentials*** - indica o código e grupo id do usuário que iniciou o processo. Isto pode ser usado para verificar a política de segurança do sistema.
- ***CPU Utilization*** - indica o tempo que o processo está executando, seja em modo usuário ou executando no núcleo do sistema.
- ***Session*** – os processos também podem ser agrupados por sessões, por exemplo, processos que pertencem aos usuários. Este campo é utilizado para armazenar informações sobre a origem de sessão que este processo pertencente.
- ***PID*** – ***P****rocess* ***ID***. Cada processo possui um identificador numérico único no sistema.
- ***Signal Support*** – os sinais são uma forma de um processo poder ser informado sobre determinado evento, por exemplo, quando o sistema está para ser finalizado. Este é um ponteiro para uma estrutura que indica como os sinais estão sendo manipulados.
- ***/proc Support*** - processos são representados no sistema de arquivos como arquivos no diretório **/proc**. O nome do arquivo deste processo é o PID.
- ***Thread List*** - um ponteiro para uma estrutura que mantém a lista de serviços do usuário que compõem este processo.
- ***Pooling*** - armazena informações sobre comportamento do processo, tais como, o estado de monitoramento do processo e o uso de recursos.
- ***Resources*** – armazena informações sobre quais recursos estão em uso e quais serão disponibilizados para o processo.
- ***Open File List*** - lista de arquivos que o processo está utilizando.

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.