

The image features a 3D maze with orange walls and a sunset background. A terminal window in the top right corner displays a green cursor and the text "#! hack'n - rio".

#! hack'n - rio

Gerenciamento de memória virtual no Kernel Linux

“conceitos básicos”

João Eriberto Mota Filho

Rio de Janeiro, RJ, 03 dez. 2011

Sumário

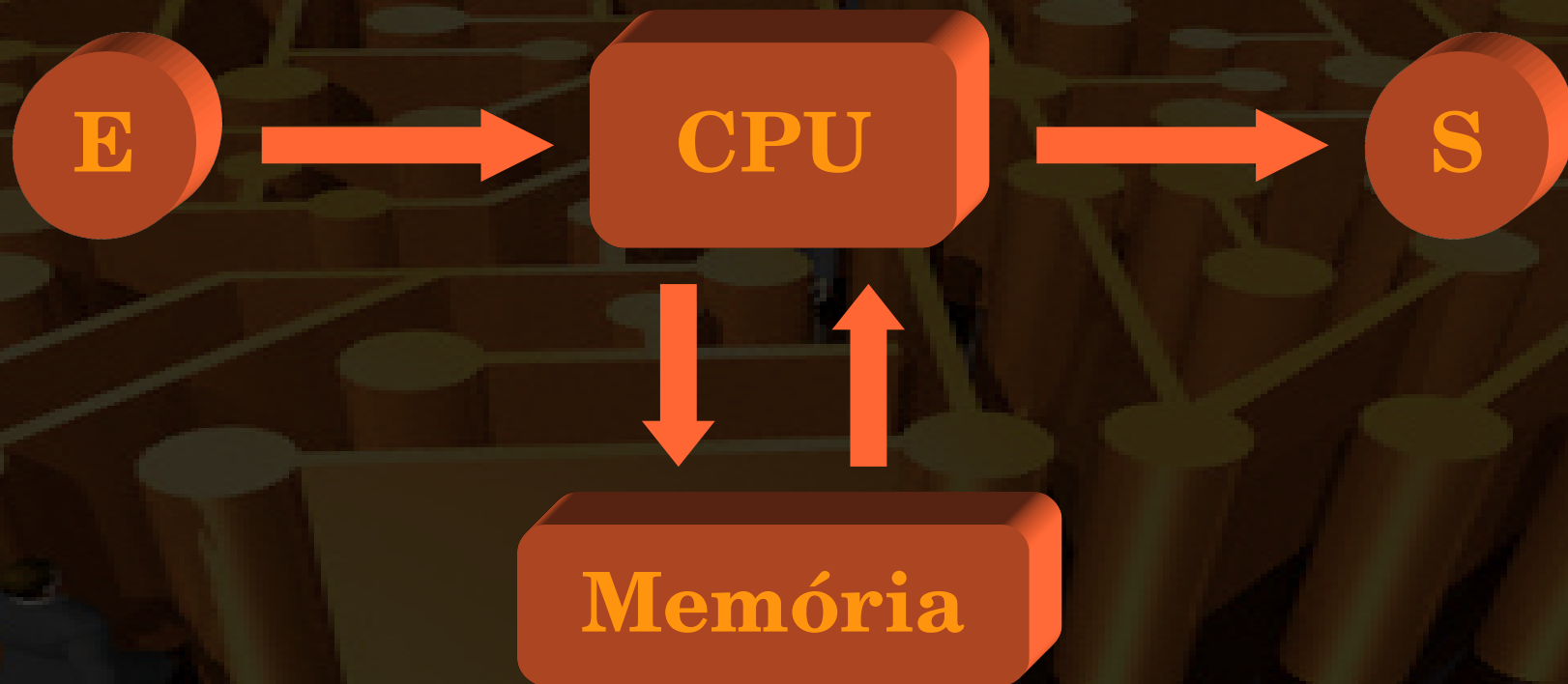
- ✓ Modelo von Neumann
- ✓ Causas de esgotamento da memória RAM
- ✓ Memória virtual e uso do swap
- ✓ Sistema buffer-cache
- ✓ Gerência do uso de memória
- ✓ Tamanhos mínimos e máximo do swap
- ✓ Swap em arquivo versus swap em partição
- ✓ O mito dos 4 GB
- ✓ Ferramentas de análise de memória
- ✓ Conclusão

Sumário

- ✓ **Modelo von Neumann**
- ✓ Causas de esgotamento da memória RAM
- ✓ Memória virtual e uso do swap
- ✓ Sistema buffer-cache
- ✓ Gerência do uso de memória
- ✓ Tamanhos mínimos e máximo do swap
- ✓ Swap em arquivo versus swap em partição
- ✓ O mito dos 4 GB
- ✓ Ferramentas de análise de memória
- ✓ Conclusão

Modelo von Neumann


- ✓ Modelo clássico de arquitetura de processamento.
- ✓ Criado por John von Neumann, em 1945.
- ✓ É uma imitação da realidade do cérebro humano.



Sumário

- ✓ Modelo von Neumann
- ✓ **Causas de esgotamento da memória RAM**
- ✓ Memória virtual e uso do swap
- ✓ Sistema buffer-cache
- ✓ Gerência do uso de memória
- ✓ Tamanhos mínimos e máximo do swap
- ✓ Swap em arquivo versus swap em partição
- ✓ O mito dos 4 GB
- ✓ Ferramentas de análise de memória
- ✓ Conclusão

Causas de esgotamento da memória RAM

- ✓ Tamanho de todos os processos em execução maior do que a quantidade de RAM existente.
- ✓ Quantidade excessiva de dados gerados por aplicações.
- ✓ Memory leak.
- ✓ Processador com baixo desempenho. 

Sumário

- ✓ Modelo von Neumann
- ✓ Causas de esgotamento da memória RAM
- ✓ **Memória virtual e uso do swap**
- ✓ Sistema buffer-cache
- ✓ Gerência do uso de memória
- ✓ Tamanhos mínimos e máximo do swap
- ✓ Swap em arquivo versus swap em partição
- ✓ O mito dos 4 GB
- ✓ Ferramentas de análise de memória
- ✓ Conclusão

Memória virtual e uso do swap

- ✓ Memória virtual: RAM + swap.
- ✓ A memória virtual dá aos programas a impressão de que há mais RAM do que realmente há.
- ✓ Programas só podem ser executados em RAM (e lá serão processos).
- ✓ Por falta de RAM, processos poderão ser passados para disco (swap), ficando na situação de espera.
- ✓ A falta de RAM leva ao uso do swap!!! Evite isso!!!
- ✓ O swap também é utilizado por sistemas operacionais que estejam com o mecanismo de hibernação ativado.

Sumário

- ✓ Modelo von Neumann
- ✓ Causas de esgotamento da memória RAM
- ✓ Memória virtual e uso do swap
- ✓ **Sistema buffer-cache**
- ✓ Gerência do uso de memória
- ✓ Tamanhos mínimos e máximo do swap
- ✓ Swap em arquivo versus swap em partição
- ✓ O mito dos 4 GB
- ✓ Ferramentas de análise de memória
- ✓ Conclusão

Sistema buffer-cache

- ✓ O sistema buffer-cache é dividido em buffer e cache.
- ✓ O buffer armazena em RAM o posicionamento nos discos de arquivos já acessados.
- ✓ O cache armazena em RAM os arquivos acessados.
- ✓ O objetivo do buffer-cache é acelerar a execução de programas e a leitura de dados.
- ✓ O buffer-cache não é essencial e pode ser esvaziado caso haja a necessidade de disponibilização de memória RAM.

Sistema buffer-cache

✓ O comando *free*:

```
cygnus:~# free -m
```

	total	used	free	shared	buffers	cached
Mem:	2016	1274	741	0	304	598
-/+ buffers/cache:		372	1643			
Swap:	399	0	399			

- ✓ O *total* é igual à RAM física menos memória de vídeo e kernel.
- ✓ A linha *Mem* representa a memória utilizada por processos, dados e buffer-cache.
- ✓ A linha *-/+ buffers/cache* mostra a quantidade real de RAM utilizada e livre (não considera o buffer-cache).
- ✓ No caso mostrado, a máquina está utilizando 372 MB RAM.

Sumário

- ✓ Modelo von Neumann
- ✓ Causas de esgotamento da memória RAM
- ✓ Memória virtual e uso do swap
- ✓ Sistema buffer-cache
- ✓ **Gerência do uso de memória**
- ✓ Tamanhos mínimos e máximo do swap
- ✓ Swap em arquivo versus swap em partição
- ✓ O mito dos 4 GB
- ✓ Ferramentas de análise de memória
- ✓ Conclusão

Gerência do uso de memória

- ✓ Programas e dados precisam de RAM para serem executados e lidos, respectivamente.
- ✓ Ao fim da execução de um programa ou leitura de disco, o buffer-cache permanece “carregado”.
- ✓ Se houver necessidade, o buffer-cache será total ou parcialmente “descarregado”.
- ✓ Se ainda houver falta de RAM, alguns dados serão transferidos para o swap.
- ✓ Para ser executado novamente (time sharing), um processo em swap deverá ser copiado de volta para a RAM.
- ✓ O swap manterá os dados até que os processos que os utilizem sejam terminados.

Gerência do uso de memória

- ✓ Processos do kernel (entre colchetes) não são “swapados”.
- ✓ Se acabar a memória virtual disponível, o processo que mais estiver onerando a memória será encerrado pelo kernel. O processo de análise poderá demorar muito.
- ✓ No fim, quase sempre, em caso de uso do swap, o mesmo ficará marcado pelos processos ainda em execução.
- ✓ DEMONSTRAÇÃO 1 - O buffer-cache em ação.
- ✓ DEMONSTRAÇÃO 2 - O uso do swap.

Sumário

- ✓ Modelo von Neumann
- ✓ Causas de esgotamento da memória RAM
- ✓ Memória virtual e uso do swap
- ✓ Sistema buffer-cache
- ✓ Gerência do uso de memória
- ✓ **Tamanhos mínimos e máximo do swap**
- ✓ Swap em arquivo versus swap em partição
- ✓ O mito dos 4 GB
- ✓ Ferramentas de análise de memória
- ✓ Conclusão

Tamanhos mínimos e máximo de swap

- ✓ Swap = 2X RAM. MITO!!! Afirmação ligada a fatores de uma época na qual memória era muito cara.
- ✓ O swap ideal: é o que não ocupa disco em excesso e dá “conforto” ao sistema operacional.
- ✓ Se o sistema de hibernação estiver ativado, o swap mínimo será o tamanho da RAM + a quantidade necessária para uma operação normal e confortável.
- ✓ O swap máximo: nos kernels 2.6 e 3.1 poderemos ter até 32 áreas de 64 GB cada (o tamanho excedente será ignorado).

Sumário

- ✓ Modelo von Neumann
- ✓ Causas de esgotamento da memória RAM
- ✓ Memória virtual e uso do swap
- ✓ Sistema buffer-cache
- ✓ Gerência do uso de memória
- ✓ Tamanhos mínimos e máximo do swap
- ✓ **Swap em arquivo versus swap em partição**
- ✓ O mito dos 4 GB
- ✓ Ferramentas de análise de memória
- ✓ Conclusão

Swap em arquivo versus swap em partição

- ✓ A partir do kernel 2.6, o swap em arquivo passou a ter o mesmo desempenho do swap em partição.
- ✓ O swap em arquivo tem diversas vantagens em relação ao swap em partição. Algumas:
 - A redução da quantidade de partições em disco (há limitações de partições e o ideal é usar somente as primárias).
 - A possibilidade de aumentar ou diminuir, rápida e facilmente, a área de swap.
 - A possibilidade de gerar, de forma simples e on-line, diversas áreas de swap.
- ✓ Uma desvantagem:
 - A possibilidade de provocar a fragmentação de arquivos. Isso só ocorrerá se o arquivo de swap for criado em uma partição de disco com pouco espaço livre (Ex.: swap de 500 MB em partição com 600 MB livres).

Swap em arquivo versus swap em partição

- ✓ Crie o arquivo de swap logo depois de instalar o sistema operacional e antes de inserir outros arquivos!
- ✓ Para saber como criar uma ou mais áreas de swap em arquivo, consulte o site http://bit.ly/swap_arquivo.

Sumário

- ✓ Modelo von Neumann
- ✓ Causas de esgotamento da memória RAM
- ✓ Memória virtual e uso do swap
- ✓ Sistema buffer-cache
- ✓ Gerência do uso de memória
- ✓ Tamanhos mínimos e máximo do swap
- ✓ Swap em arquivo versus swap em partição
- ✓ **O mito dos 4 GB**
- ✓ Ferramentas de análise de memória
- ✓ Conclusão

O mito dos 4 GB

- ✓ Uso kernel de 32 bits e, apesar de ter 4 GB RAM, somente 3 GB são mostrados. Por quê? O que fazer?
- ✓ As arquiteturas de 32 bits endereçam até 4 GB RAM (2^{32} conjuntos de 8 bits de transmissão de dados ou bytes).
- ✓ O espaço do usuário é limitado a 3 GB em arquiteturas de 32 bits (i386). Isso faz parte de um padrão antigo.
- ✓ A solução é utilizar um kernel de 32 bits com PAE (Physical Address Extension) habilitado ou um kernel de 64 bits.
- ✓ Os kernels com PAE endereçam até 64 GB de RAM (2^{36}).
- ✓ Kernels de 64 bits, teoricamente, endereçam até 2^{64} bytes.
- ✓ Na prática, o atual Kernel Linux de 64 bits endereça 2^{46} bytes (64 TB).

Sumário

- ✓ Modelo von Neumann
- ✓ Causas de esgotamento da memória RAM
- ✓ Memória virtual e uso do swap
- ✓ Sistema buffer-cache
- ✓ Gerência do uso de memória
- ✓ Tamanhos mínimos e máximo do swap
- ✓ Swap em arquivo versus swap em partição
- ✓ O mito dos 4 GB
- ✓ **Ferramentas de análise de memória**
- ✓ Conclusão

Ferramentas de análise de memória

- ✓ Comando *free*: mostra o uso de memória virtual.
- ✓ Comando *memstat*: mostra o uso da memória por parte dos diferentes processos. Útil para saber qual processo está consumindo memória de forma exagerada.
- ✓ Comando *pmap*: mostra, detalhadamente, como um determinado processo (citado pelo seu PID) está utilizando a memória, incluindo outros processos ligados a ele.
- ✓ Ambiente *top*: mostra, interativamente, dentre outras coisas, como os processos ocupam a memória.
- ✓ MRTG: cria gráficos, inclusive da memória, facilitando a análise de uso de RAM e swap por períodos de tempo. Muito útil para ajustes em máquinas virtuais.

Ferramentas de análise de memória

✓ Comando memstat:

```
cygnus:~# memstat | sort -nr
```

```
1038740k ( 430288k)
 96692k: PID 2253 (/usr/lib/openoffice/program/soffice.bin)
 78376k: PID 1955 (/usr/bin/knotify4)
 32824k( 0k): /usr/lib/locale/locale-archive 1051 1336 1915 1916 1918
 29392k: PID 1953 (/usr/bin/plasma-desktop)
 28900k: PID 1196 (/usr/bin/Xorg)
 25864k: PID 3996 (/usr/bin/sort)
 25784k: PID 1051 (/usr/lib/speech-dispatcher-modules/sd_espeak)
 25132k: PID 1052 (/usr/sbin/rsyslogd)
 20708k( 15844k): /usr/lib/libQtWebKit.so.4.6.3 1915 1916 1918 1945 1948
 20608k: PID 1979 (/usr/bin/kmix)
 19012k: PID 1396 (/usr/bin/python2.6)
 15860k( 15860k): /var/tmp/kdecache-eriberto/kpc/kde-icon-cache.data 1918
 15640k( 0k): /usr/lib/libicudata.so.42.1 2253
 13548k: PID 1978 (/usr/bin/krunner)
```

```
...
```

Ferramentas de análise de memória

✓ Comando pmap:

```
cygnus:~# pmap -x 2253
```

```
2253: /usr/lib/openoffice/program/soffice.bin -impress -splash-pipe=5
```

Address	Kbytes	RSS	Dirty	Mode	Mapping
08048000	0	8	0	r-x--	soffice.bin
0804a000	0	4	4	rw---	soffice.bin
08d45000	0	49556	49556	rw---	[anon]
ab177000	0	1200	1200	rw---	[anon]
ab2a3000	0	272	0	r-x--	libfwlli.so
ab314000	0	16	16	rw---	libfwlli.so
ab318000	0	568	0	r-x--	libcuili.so
ab6cb000	0	112	112	rw---	libcuili.so
ab6e7000	0	4	4	rw---	[anon]
ab6e8000	0	144	0	r-x--	libsduili.so
ab775000	0	20	20	rw---	libsduili.so
ab77a000	0	60	0	r--s-	DejaVuSansMono.ttf
ab7c9000	0	20	0	r--s-	crystalr.ttf
ab7d1000	0	48	0	r--s-	creditri.ttf

```
...
```

Ferramentas de análise de memória

✓ Ambiente top:

```
top - 17:35:38 up 2:16, 2 users, load average: 0.32, 0.28, 0.14
Tasks: 114 total, 2 running, 112 sleeping, 0 stopped, 0 zombie
Cpu(s): 21.1%us, 27.7%sy, 0.0%ni, 49.8%id, 0.0%wa, 0.7%hi, 0.7%si,
0.0%st
Mem: 2064472k total, 773744k used, 1290728k free, 127904k buffers
Swap: 409592k total, 0k used, 409592k free, 388624k cached
```

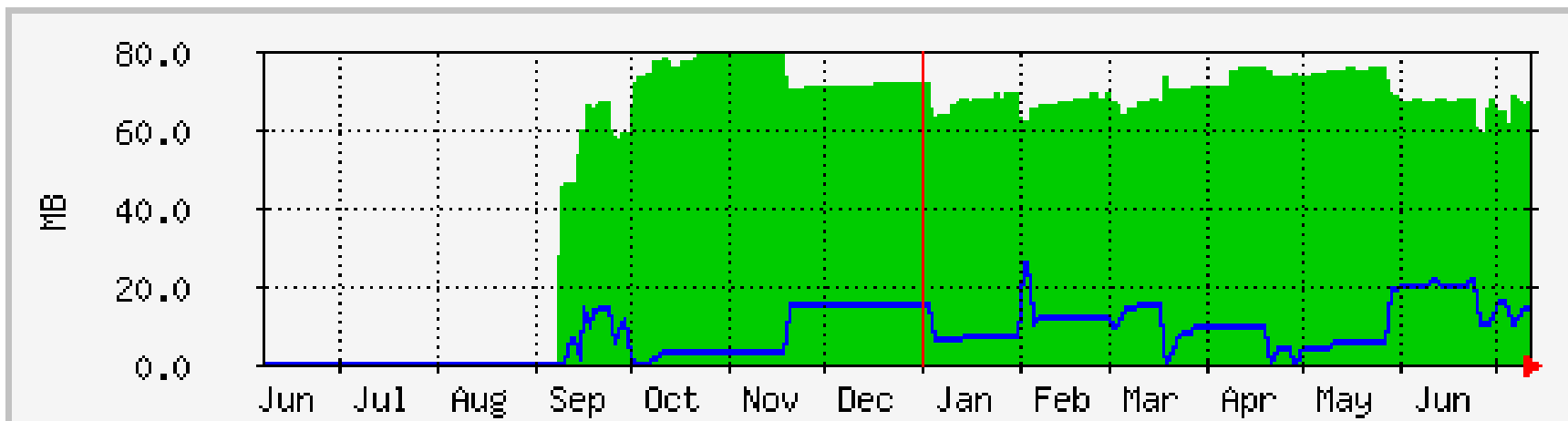
PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2253	eriberto	20	0	247m	127m	69m	S	19.2	6.3	5:28.07	soffice.bin
1953	eriberto	20	0	281m	39m	24m	S	1.0	1.9	2:05.91	plasma-desktop
1948	eriberto	20	0	252m	28m	21m	S	0.7	1.4	0:59.41	kwin
1955	eriberto	20	0	161m	28m	15m	S	0.0	1.4	0:02.15	knotify4
1978	eriberto	20	0	269m	27m	20m	S	0.0	1.4	0:02.10	krunner
1979	eriberto	20	0	263m	21m	15m	S	0.0	1.1	0:00.95	kmix
1918	eriberto	20	0	108m	19m	14m	S	0.0	1.0	0:01.66	kded4
1985	eriberto	20	0	68992	18m	14m	S	0.0	0.9	0:00.84	knetworkman
1984	eriberto	20	0	78828	18m	13m	S	0.0	0.9	0:00.66	kgpg

...

Ferramentas de análise de memória

✓ MRTG (em máquina virtual Xen):

Gráfico 'Anual' (1 dia Média)



	Máx	Média	Atual
RAM	79.0 MB (7.7%)	70.0 MB (6.8%)	67.0 MB (6.5%)
Swap	25.0 MB (2.4%)	10.0 MB (1.0%)	14.0 MB (1.4%)

Sumário

- ✓ Modelo von Neumann
- ✓ Causas de esgotamento da memória RAM
- ✓ Memória virtual e uso do swap
- ✓ Sistema buffer-cache
- ✓ Gerência do uso de memória
- ✓ Tamanhos mínimos e máximo do swap
- ✓ Swap em arquivo versus swap em partição
- ✓ O mito dos 4 GB
- ✓ Ferramentas de análise de memória
- ✓ **Conclusão**

Conclusão

- ✓ Processos em execução devem estar na RAM.
- ✓ Falta de RAM: pode ser baixo desempenho do processador.
- ✓ O uso de swap é um indicativo de falta de memória RAM ou de hibernação. Elementos do kernel não são swapados!
- ✓ No comando `free`, é a linha `-/+ buffers/cache` quem mostra a real utilização da RAM.
- ✓ O swap não é o dobro da RAM!
- ✓ Para 4 GB RAM ou mais, use PAE ou um kernel de 64 bits.

Esta palestra está disponível em:

<http://eriberto.pro.br>

Siga-me em <http://twitter.com/eribertomota>