III Forum Amazonico de Software Livre

Conceitos sobre memória virtual para análise forense computacional

João Eriberto Mota Filho

Santarém, PA, 31 de agosto de 2011

Sumário

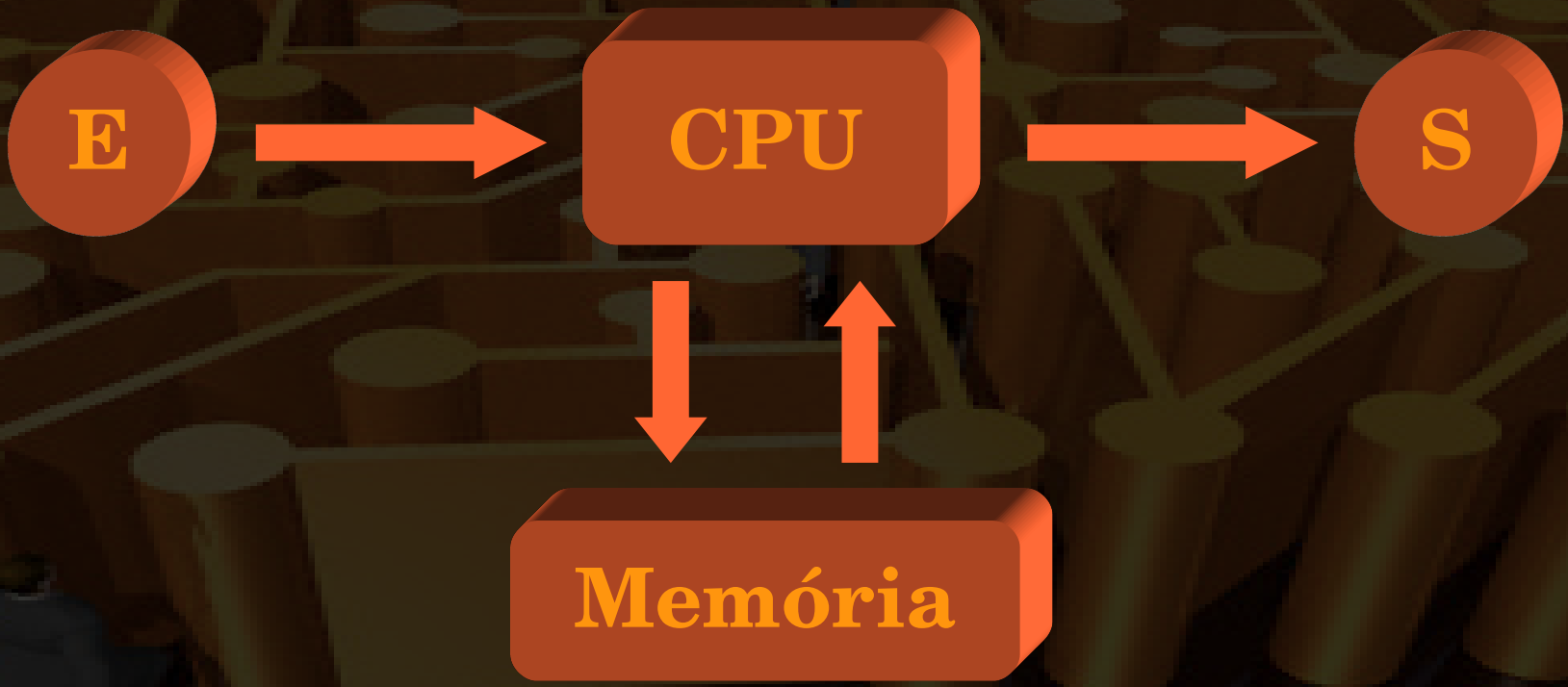
- ✔ Modelo von Neumann
- ✔ Causas de esgotamento da memória RAM
- ✔ Memória virtual e uso do swap
- ✔ Sistema buffer-cache
- ✔ Gerência do uso de memória
- ✔ Dump de memória
- ✔ Conclusão

Sumário

- ✔ Modelo von Neumann
- ✔ Causas de esgotamento da memória RAM
- ✔ Memória virtual e uso do swap
- ✔ Sistema buffer-cache
- ✔ Gerência do uso de memória
- ✔ Dump 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
- ✔ Dump 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
- ✔ Dump 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.
- ✔ O swap poderá ser implementado em partições ou em arquivos (melhor opção atualmente).

Sumário

- ✔ Modelo von Neumann
- ✔ Causas de esgotamento da memória RAM
- ✔ Memória virtual e uso do swap
- ✔ **Sistema buffer-cache**
- ✔ Gerência do uso de memória
- ✔ Dump 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.
- ✔ Graças ao buffer-cache, os dados utilizados permanecem na RAM depois de utilizados (podendo ser sobrescritos).

Sistema buffer-cache

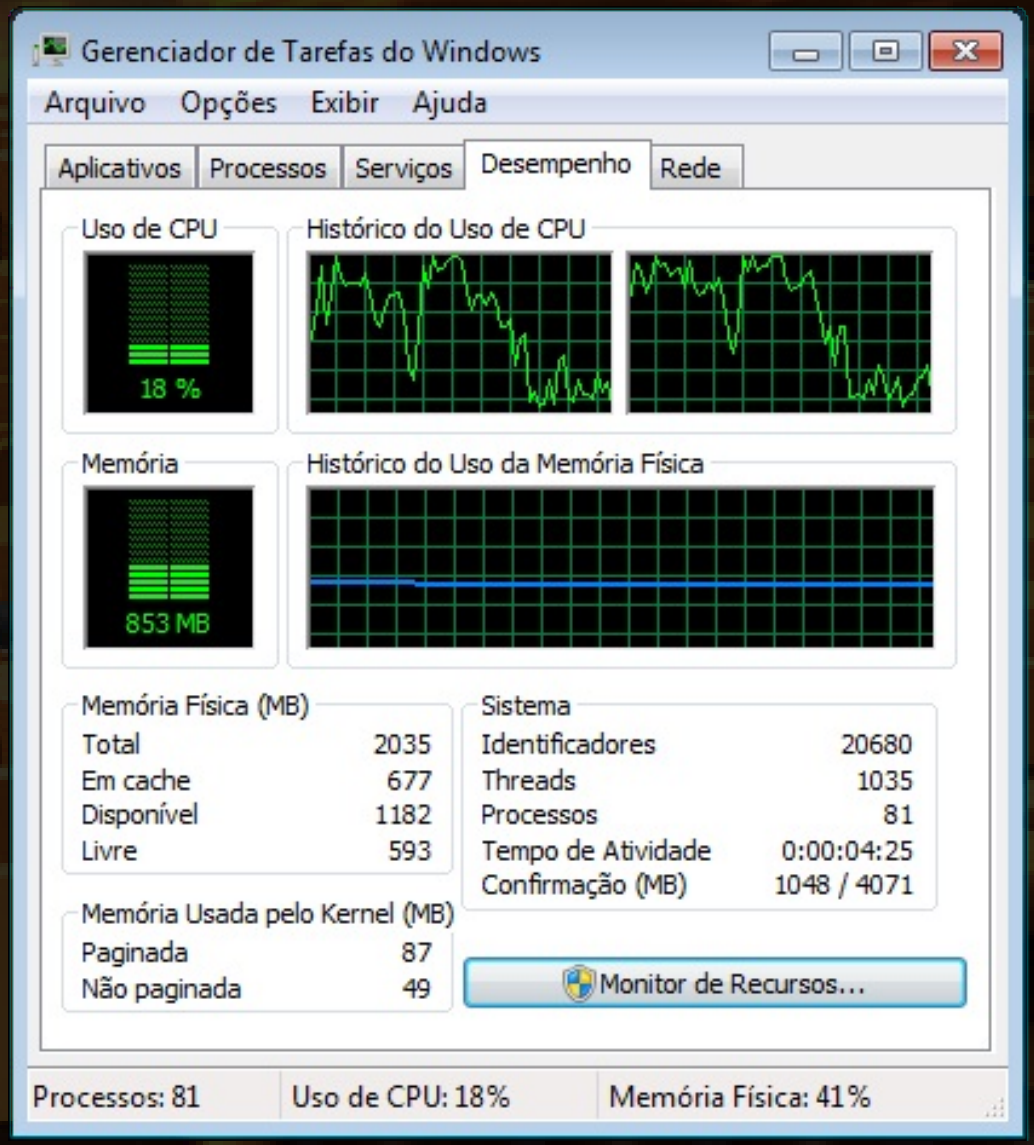
- O comando *free* no GNU/Linux:

```
root@antares:~# free -m
             total       used       free     shared    buffers     cached
Mem:          2001       1032        968          0         37        608
-/+ buffers/cache:        386       1614
Swap:          499          0        499
```

- 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 386 MB RAM.

Sistema buffer-cache

- ✔ No MS Windows é possível observar o uso da memória por intermédio do gerenciador de tarefas.



Sumário

- ✔ Modelo von Neumann
- ✔ Causas de esgotamento da memória RAM
- ✔ Memória virtual e uso do swap
- ✔ Sistema buffer-cache
- ✔ **Gerência do uso de memória**
- ✔ Dump 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.

Sumário

- ✔ Modelo von Neumann
- ✔ Causas de esgotamento da memória RAM
- ✔ Memória virtual e uso do swap
- ✔ Sistema buffer-cache
- ✔ Gerência do uso de memória
- ✔ **Dump de memória**
- ✔ Conclusão

Dump de memória

- ✔ O dump de memória no GNU/Linux poderá ser feito com o auxílio do LKM fmem, pertencente ao projeto Foriana.
- ✔ Será gerado um dispositivo /dev/fmem que dará o acesso à memória por intermédio do comando dd ou dcfldd ou similar. Exemplo (para 4 GB RAM):

```
# dcfldd if=/dev/fmem of=memo.dd bs=1M count=4096
```

- ✔ No MS Windows há programas que realizam o dump de memória, como o win32dd, todos disponíveis na suíte WinTaylor, que é parte do projeto CAINE.

Sumário

- ✔ Modelo von Neumann
- ✔ Causas de esgotamento da memória RAM
- ✔ Memória virtual e uso do swap
- ✔ Sistema buffer-cache
- ✔ Gerência do uso de memória
- ✔ Dump de memória
- ✔ Conclusão

Conclusão

- ✔ Processos em execução devem estar na RAM.
- ✔ Tudo passa pela memória!
- ✔ O uso de swap é um indicativo de falta de memória RAM ou de hibernação. Elementos do kernel não são swapados!
- ✔ A operação de dump de memória implica na perda de uma pequena parte das informações nela armazenadas.

Esta apresentação está baseada na palestra
"Gerenciamento de memória virtual no Kernel Linux".
Ambas estão disponíveis em:

http://www.eriberto.pro.br

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