

Atividade 4

Gerência de Memória Principal

1. Experiências Práticas:

(a) Windows (Sysinternals Tools):

Executado o ProcessExplorer.

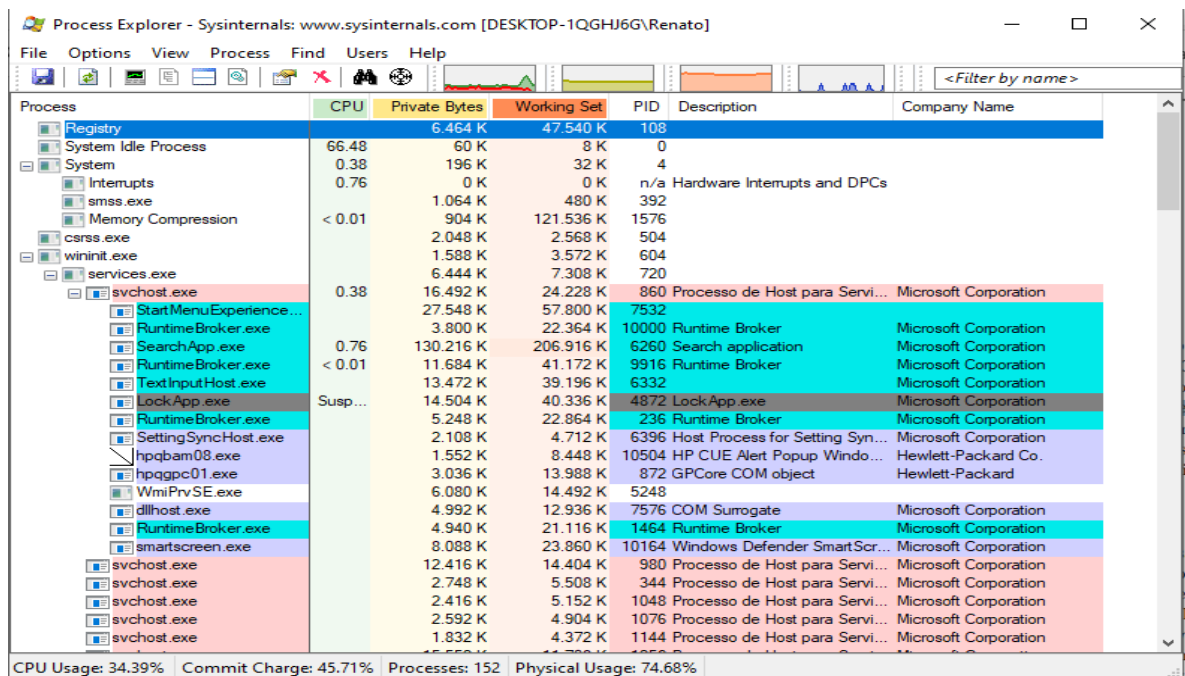


Figura 1: Execução do Process Explorer

Após clicar no botão System Information, foi exibido os seguintes resultados:

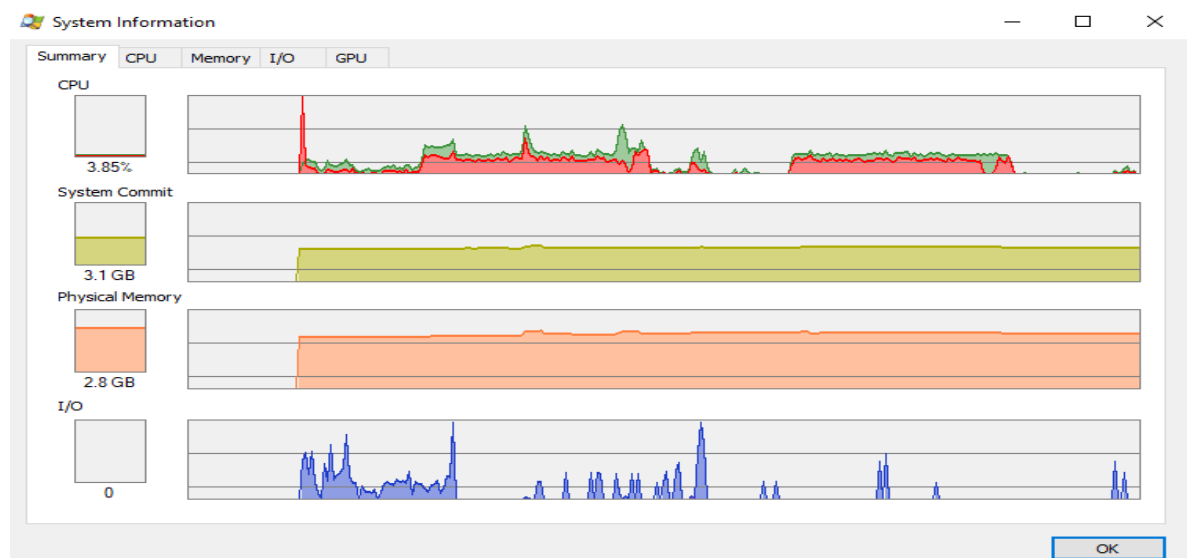


Figura 2: System Information

Estava sendo utilizado 2.8GB da memória física do sistema com a taxa de uso da CPU em 3.85%. Sobre a paginação e memória kernel, os resultados foram os obtidos abaixo:

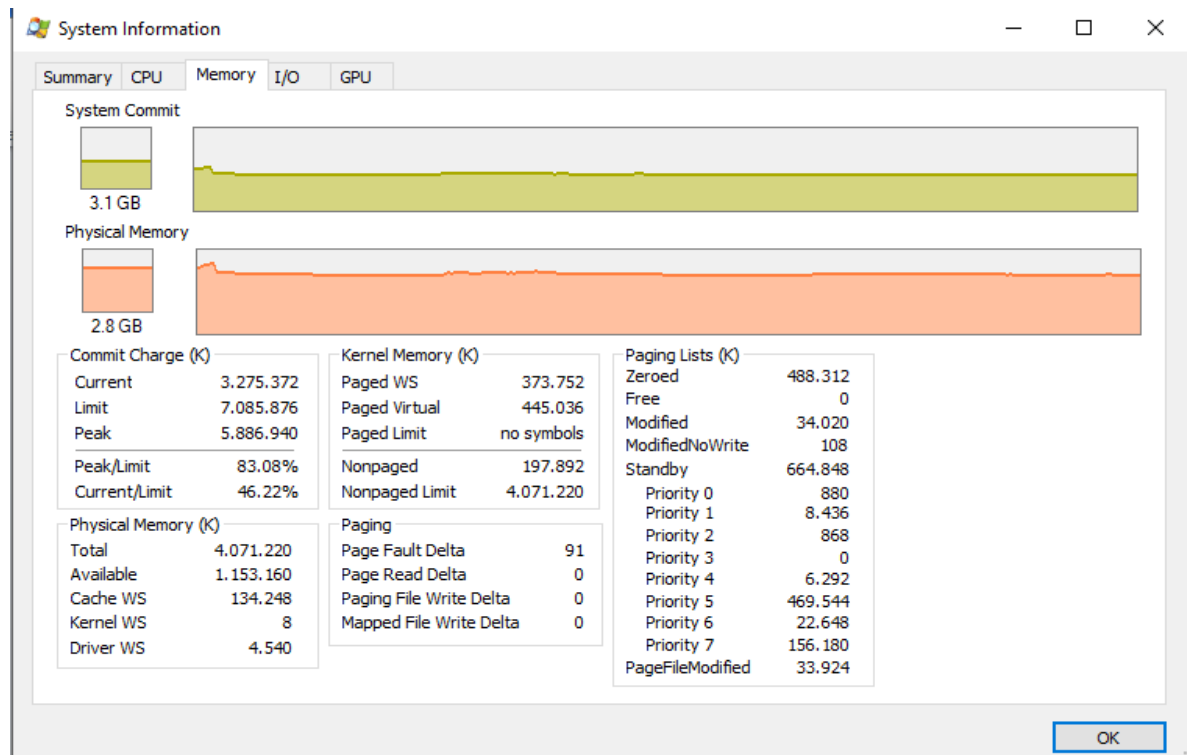


Figura 3: System Information – Paginação e memória

Na memória física do total de 4GB, está disponível 1,153GB, com o uso de 2.7 a 2.8GB. Na memória do commit charge que deve ser respaldada pelos processos, está com a quantia de 3,275GB, com pico de 5,8GB e o limite é cerca de 7GB. Sobre a memória kernel, a paginação virtual tá em 445MB enquanto a WS tá em 373 MB. As não paginadas são 197MB.

Sobre as listas de paginação, as paginações zeradas atingiu 488MB, que são páginas zeradas de dados armazenadas anteriormente que foram preenchidas com zero sendo um recurso de segurança. As paginações modificadas 34MB e a espera de 664MB.

Foi iniciado agora, um programa novo para análise novamente, para isso foi aberto o jogo League of Legends para um comparativo. Após a abertura do jogo os resultados foram estes abaixo:

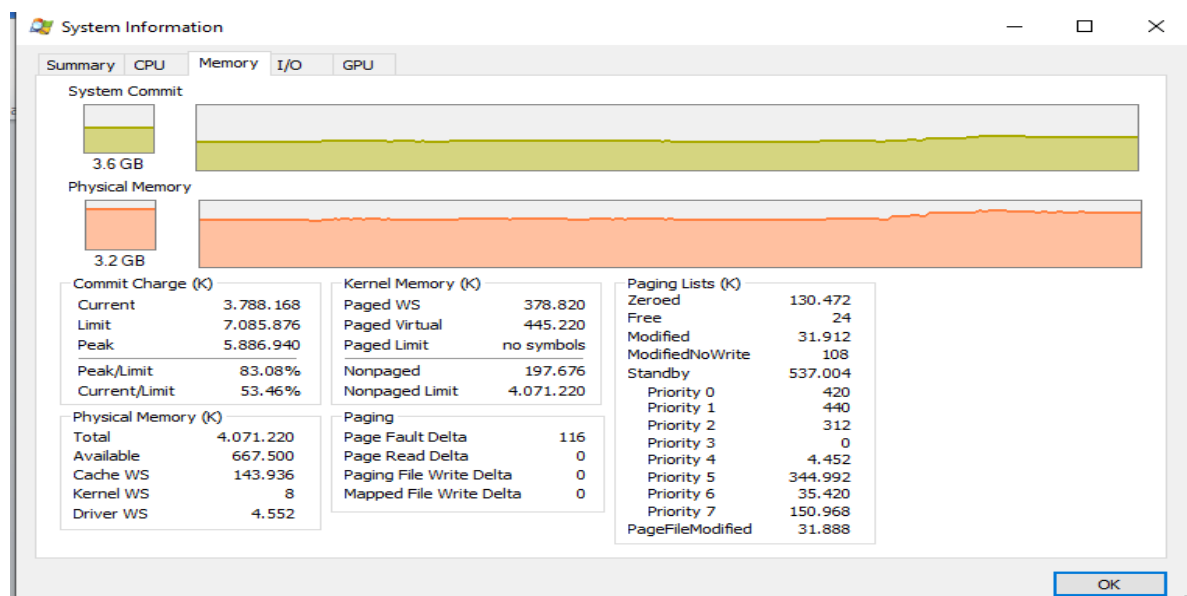


Figura 4: System Information – Paginação e memória após novo programa

Foi visto que o uso da memória física aumentou drasticamente, antes de 2.7 GB saltou para 3.2GB. A memória do commit charge de processos, que é virtual, foi de 3,2GB anteriormente para 3,7GB. A memória kernel WS foi de 373MB para 375MB, não sofrendo tanta alteração e a memória kernel página virtual manteve parecida. As paginações zeradas abaixaram de 488MB para 130MB. A taxa de uso de CPU aumentou muito também antes de 3% foi para 33%.



Figura 5: Taxa Cpu com o novo programa rodando

Um minuto após fechar o jogo League of Legends que foi aberto para a análise, os resultados voltaram parecidos com a etapa anterior antes da execução, com uso de CPU menor, memória física menor voltando para 2,7GB. Memória virtual de system commit em 3,2GB novamente, diminuiu fechando o programa e a paginação zeroed (zerada) voltou com o índice maior após fechar o programa. Porém agora, tem a paginação livre (FREE) que antes de iniciar o programa estava quase zerada e após a execução foi para cerca de 31MB.

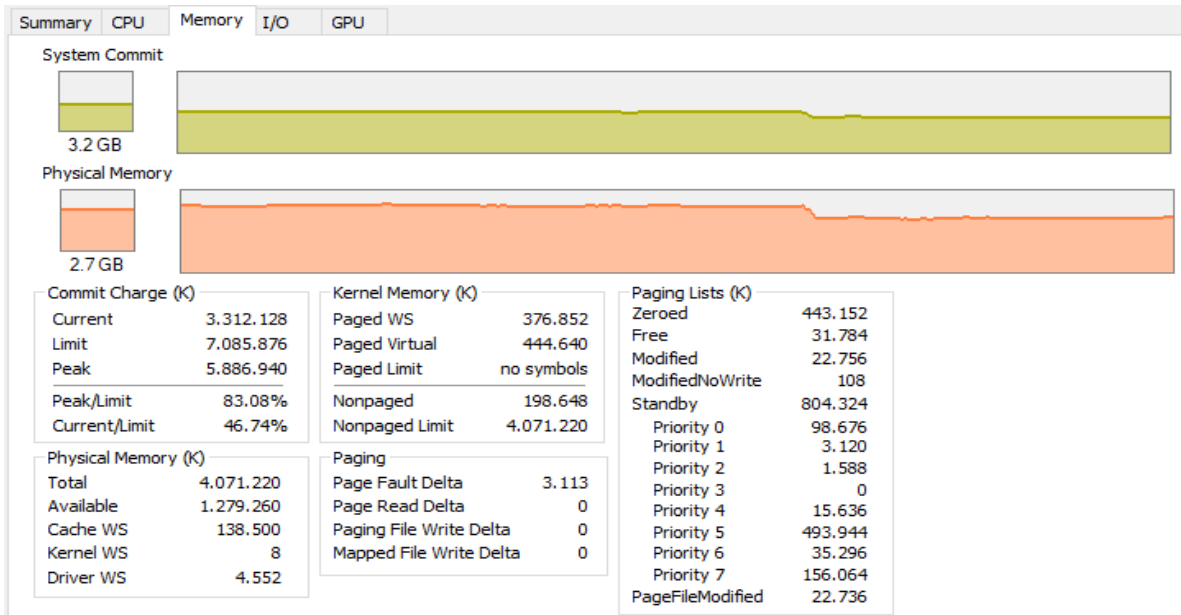


Figura 6: Resultado após fechar programa

(b) Linux:

O comando free é útil para observação da memória do sistema.

Após executar o comando man free, obtive o manual com explicações sobre sua utilização.

```

FREE(1)                                     User Commands                                     FREE(1)

NAME
    free - Display amount of free and used memory in the system

SYNOPSIS
    free [options]

DESCRIPTION
    free displays the total amount of free and used physical and swap mem-
    ory in the system, as well as the buffers and caches used by the ker-
    nel. The information is gathered by parsing /proc/meminfo. The dis-
    played columns are:

    total    Total installed memory (MemTotal and SwapTotal in /proc/meminfo)
    used     Used memory (calculated as total - free - buffers - cache)
    free     Unused memory (MemFree and SwapFree in /proc/meminfo)
    shared   Memory used (mostly) by tmpfs (Shmem in /proc/meminfo)
    buffers  Buffers used by the kernel (Bmem in /proc/meminfo)

Manual page free(1) line 1 (press h for help or q to quit)

```

Figura 7: Comando man free

Após a execução do free algumas vezes durante a execução de outros programas em um certo intervalo de tempo por cerca de uns 20 segundos para análise da mudança da memória, foi obtido as seguintes sequências de resultados abaixo:

	total	used	free	shared	buff/cache	available
Mem:	32884784	2175464	11485588	4324	19223732	28678484
Swap:	0	0	0			

Figura 8: Resultado de execução do comando free primeira vez

	total	used	free	shared	buff/cache	available
Mem:	32884784	2184068	11474976	4332	19225740	28671920
Swap:	0	0	0			

Figura 9: Resultado de execução do comando free segunda vez

	total	used	free	shared	buff/cache	available
Mem:	32884784	2191016	11467456	4344	19226312	28664296
Swap:	0	0	0			

Figura 10: Resultado de execução do comando free terceira vez

	total	used	free	shared	buff/cache	available
Mem:	32884784	2210676	11447840	4344	19226268	28644636

Figura 11: Resultado de execução do comando free quarta vez

Analisando os dados, foi visto que na primeira vez com nenhum programa aberto praticamente, o uso da memória foi menor e a memória livre é maior e um índice menor de compartilhada. A memória disponível foi maior no primeiro caso visto que nenhum programa estava rodando. Quando acrescentado os programas em execução, a memória disponível foi diminuindo consideravelmente, a memória disponível nada mais é do que a memória utilizável onde a memória livre e a memória de buffer cache podem ser utilizadas. Com isso ela se tornou menor com os programas executando e o número de memória used(usada) foi aumentando devido os programas e claro a memória livre diminuindo a cada print.

Usado o comando free -m no último print, para termos os resultados em MB.

Eixo X -> MB usado = 21Mb pelo programa libreoffice aberto

Eixo Y -> Mem usada=2210MB em 20 segundos executados.

Eixo X -> MB usado = 21Mb pelo programa libreoffice aberto

Eixo Y -> Mem cache=1922MB em 20 segundos executados.

Eixo X -> MB usado = 21Mb pelo programa libreoffice aberto

Eixo Y -> paginacao=1088MB em 20 segundos executados.

A memória em cache tem o objetivo de guardar dados, informações e processos temporários acessados com frequência para maior agilidade durante o processo de uso quando o usuário executa, sendo assim o seu valor depende da quantidade dos programas abertos no momento.