

DOCUMENTAÇÃO RESULTADOS EXPERIMENTAIS

Experimento 1: Overhead de Monitoramento

```
rfafe@Rafael1:/mnt/c/Users/rfafe/OneDrive/Documentos/GitHub/resource-monitor$ ./resource-monitor -o 100
Calculando Overhead de namespaces
Iteracoes: 100

[1] Tempo total (fork, baseline): 37.33 ms
[2] Tempo total (clone + CLONE_NEWPID): 6.39 ms

-----
Overhead de Namespace
Diferenca de tempo total: -30.94 ms
Overhead medio por criacao: -309.409 us
```

O Overhead de monitoramento calcula o custo de tempo extra que o kernel Linux demora para a criação de um processo ou namespace PID novos comparando a a criação de um processo comum.

O primeiro tempo é reporta o tempo base (fork), que é a ação do processo pai sendo duplicado pelo kernel sem a criação de um novo namespace, e os namespaces existentes são herdados pelo processo filho.

O segundo tempo relata a criação de um novo processo pelo Kernel e simultaneamente cria um novo namespace PID isolado para ele.

A diferença de tempo total é a relação entre o primeiro tempo e o segundo, a causa do resultado negativo demonstra que criar um processo com um novo namespace PID é significativamente mais rápido do que criar um processo sem um namespace novo.

Experimento 2: Isolamento via Namespaces

PID	COMANDO	NET_ID	PID_ID	MNT_ID
294	bash	4026531840	4026532221	4026532219
358	bash	4026531840	4026532221	4026532219
929	sh	4026531840	4026532221	4026532219
930	sh	4026531840	4026532221	4026532219
936	sh	4026531840	4026532221	4026532219
940	node	4026531840	4026532221	4026532219
953	node	4026531840	4026532221	4026532219
962	node	4026531840	4026532221	4026532219
1000	node	4026531840	4026532221	4026532219
1016	node	4026531840	4026532221	4026532219
1031	node	4026531840	4026532221	4026532219
1121	cpptools	4026531840	4026532221	4026532219
1163	node	4026531840	4026532221	4026532219
1549	pet	4026531840	4026532221	4026532219
1664	bash	4026531840	4026532221	4026532219
1745	bash	4026531840	4026532221	4026532219
2075	node	4026531840	4026532221	4026532219
2647	bash	4026531840	4026532221	4026532219
2731	node	4026531840	4026532221	4026532219
2740	node	4026531840	4026532221	4026532219
2747	node	4026531840	4026532221	4026532219
2772	node	4026531840	4026532221	4026532219
2887	cpptools	4026531840	4026532221	4026532219
2926	node	4026531840	4026532221	4026532219
2992	bash	4026531840	4026532221	4026532219
7322	bash	4026531840	4026532221	4026532219
20664	python	4026531840	4026532221	4026532219
20742	python	4026531840	4026532221	4026532219
20883	bash	4026531840	4026532221	4026532219
23351	node	4026531840	4026532221	4026532219
23407	node	4026531840	4026532221	4026532219
24017	cpptools-srv	4026531840	4026532221	4026532219
24991	cpptools-srv	4026531840	4026532221	4026532219
26607	resource-monito	4026531840	4026532221	4026532219

Relatorio de namespaces gerado com sucesso: Namespace_Report.csv

O isolamento via namespaces analisa a efetividade de isolamentos de processos por meio de combinações diferentes de namespaces. Ele cria processos com diferentes configurações de namespaces, testa os recursos do sistema, interfaces de rede e áreas do sistema de arquivo verificando o isolamento. O tempo de criação dos tipos de namespaces também são medidos para verificar o overhead da operação.

Experimento 3: Throttling de CPU

```
EXPERIMENTO 3: THROTTLING DE CPU
=====
LIMITE (cores) | CPU% MEDIDO | THROUGHPUT (iter/s) | DESVIO%
-----
CRIANDO E CONFIGURANDO CGROUP:
Controlador: cpu
Nome: throttle_0.25
CPU: 0.25 cores
Memoria: 0 MB
Cgroup ja existe: /sys/fs/cgroup/throttle_0.25
CPU configurado: max=25000us, period=100000us
Cgroup criado e configurado com sucesso
Processo 27884 movido para cgroup: /sys/fs/cgroup/throttle_0.25
0.25      | 27.8      | 126230991      | 11.1
CRIANDO E CONFIGURANDO CGROUP:
Controlador: cpu
Nome: throttle_0.50
CPU: 0.50 cores
Memoria: 0 MB
Cgroup ja existe: /sys/fs/cgroup/throttle_0.50
CPU configurado: max=50000us, period=100000us
Cgroup criado e configurado com sucesso
Processo 27884 movido para cgroup: /sys/fs/cgroup/throttle_0.50
0.50      | 69.6      | 333166073      | 39.3
CRIANDO E CONFIGURANDO CGROUP:
Controlador: cpu
Nome: throttle_1.00
CPU: 1.00 cores
Memoria: 0 MB
Cgroup ja existe: /sys/fs/cgroup/throttle_1.00
CPU configurado: max=100000us, period=100000us
Cgroup criado e configurado com sucesso
Processo 27884 movido para cgroup: /sys/fs/cgroup/throttle_1.00
1.00      | 102.4     | 499631657      | 2.4
CRIANDO E CONFIGURANDO CGROUP:
Controlador: cpu
Nome: throttle_2.00
CPU: 2.00 cores
Memoria: 0 MB
Cgroup ja existe: /sys/fs/cgroup/throttle_2.00
CPU configurado: max=200000us, period=100000us
Cgroup criado e configurado com sucesso
Processo 27884 movido para cgroup: /sys/fs/cgroup/throttle_2.00
2.00      | 100.4     | 457897116      | -49.8
-----
Experimento de throttling concluido
```

O objetivo do experimento de throttling de CPU é verificar a precisão de limitação de uso de CPU ao aplicar a restrição por meio do cgroups. Ele executa o CPU-intensive inicialmente sem as restrições de uso. Aplica os limites do CPU com a utilização do cgroups, testando diferentes configurações de cores. Mede o uso real de CPU para as configurações observando o comportamento do throughput à limitação.

Experimento 4: Limitação de memória

1. Criar cgroup com limite de 100MB
2. Tentar alocar memoria incrementalmente
3. Observar comportamento (OOM killer, falhas de alocacao)

CRIANDO E CONFIGURANDO CGROUP:

Controlador: memory

Nome: exp_memoria

CPU: 0.00 cores

Memoria: 100 MB

Cgroup ja existe: /sys/fs/cgroup/exp_memoria

Memoria configurada: 104857600 bytes

Cgroup criado e configurado com sucesso

Processo 29337 movido para cgroup: /sys/fs/cgroup/exp_memoria

TESTE DE ALOCACAO DE MEMORIA:

Tentativa | Tamanho (MB) | Status

Tentativa	Tamanho (MB)	Status
1	10	ALOCADO Memoria atual: 0.91 MB
2	25	ALOCADO Memoria atual: 0.91 MB
3	50	ALOCADO Memoria atual: 0.91 MB
4	75	ALOCADO Memoria atual: 0.91 MB
5	100	ALOCADO Memoria atual: 71.20 MB
6	110	ALOCADO Memoria atual: 0.50 MB
7	120	ALOCADO Memoria atual: 53.12 MB

Erro ao ler de /sys/fs/cgroup/exp_memoria/memory.events

RESULTADOS FINAIS:

Quantidade maxima alocada: 120 MB

Numero de falhas: 0

Contador de falhas (memory.failcnt): -1

Comportamento: Todas alocacoes bem-sucessidas

A limitação de memória visa testar comportamento de processos que possuem um limite de memória pré-determinado por um cgroup. Ele cria um cgroup com limite de memória de 100MB, realizando a tentativa de alocar memória de forma incremental (em etapas). Também observa quando o limite é atingido e registra falhas de alocação ou o possível acionamento do OOM killer.

