

Escalonamento de Processos

Tarefa Prática

Objetivo da Tarefa

- Observar como o Linux distribui CPU entre processos concorrentes
- Analisar o impacto de sobrecarga, bloqueio por E/S e prioridade
- Relacionar as observações com o funcionamento do escalonador CFS
- Comparar com os algoritmos clássicos: FIFO, RR e SJF

O Escalonador do Linux (CFS)

- CFS: Completely Fair Scheduler
- Organiza processos em uma árvore red-black
- Prioriza quem usou menos CPU recentemente (vruntime)
- Valor de `nice` ajusta o peso (mais `nice` = menos CPU)
- Foco em justiça, não em prioridades absolutas

Estados de um Processo no Linux

- Estados mais comuns

Código	Nome	Significado	Teoria
R	Running	Executando ou pronto para executar	Pronto/Executando
S	Sleeping	Esperando por um evento (ex: entrada do teclado)	Bloqueado
D	Uninterruptible Sleep	Esperando E/S longa (ex: disco)	Bloqueado
T	Stopped	Parado (ex: CTRL+Z)	Parado
Z	Zombie	Finalizado, mas não coletado pelo pai	Finalizado

Parte 1 – Distribuição com N Processos

1. Descubra o número de núcleos N usando `nproc`
2. Inicie N processos contendo um laço infinito (consumo contínuo de CPU)
3. Observe com `top`, `htop` ou `ps`
4. Anote:
 - Uso de CPU (`%CPU`, `TIME+`)
 - Estado dos processos (`R`, `S`, etc.)
5. Encerre os processos após observação

Parte 2 – Sobrecarga com $N+1$ Processos

- Repita a execução com $N+1$ processos
- Observe se a CPU é distribuída igualmente
- Há algum processo que recebe menos CPU?
- O sistema continua justo ou há preempção perceptível?

Parte 3 – Efeito da Prioridade

- Use `renice -n -10 -p <PID>` para aumentar a prioridade de um processo
- Observe o impacto no uso de CPU
- Como os outros processos reagem?
- CFS deve favorecer o processo com menor `vruntime` (mais prioridade)

Parte 4 – Processo Bloqueado por Entrada

- Execute N processos intensivos + 1 processo que aguarda entrada:

```
while true; do read var; done
```

- Observe:
 - Estado (s) e uso de CPU
 - Comportamento ao receber entrada
- Aumente a prioridade e veja se há impacto no uso de CPU

Análise Final (Relatório)

Responda:

- Como o escalonador do Linux distribuiu a CPU?
- O que aconteceu na sobrecarga?
- Qual o efeito da prioridade?
- O processo bloqueado recebeu mais CPU?
- Explique com base no CFS
- Compare com: FIFO, Round Robin, SJF

Ferramentas Sugeridas

- `top`, `htop`, `ps` para monitoramento
- `nproc` para núcleos disponíveis
- `renice`, `nice` para ajustar prioridades
- `kill`, `killall` para encerrar processos
- Comandos úteis:

```
ps -eo pid,pri,ni,stat,%cpu,cmd --sort=-%cpu
```