


| | | | | |
|---|--|-------------|-----------|-------|
|  | INSTITUTO FEDERAL DA PARAIBA - IFPB (CAMPUS JOÃO PESSOA) | | | |
| | Componente Curricular: SISTEMAS OPERACIONAIS | | | |
| | Professor: ANDRÉ LUÍS DE LUCENA TORRES | | | |
| | Semestre: 2025.1 | Período: 4º | Turma: SI | Data: |

| |
|-----------------------------|
| Assinatura do (a) Aluno(a): |
| ATIVIDADE INTEGRADORA |

🎯 Objetivo Geral:

Desenvolver um projeto prático que integre os principais conceitos teóricos e práticos abordados na disciplina de Sistemas Operacionais, aplicando-os na criação, simulação e análise de componentes de um sistema operacional com uso de software livre.

Descrição da Atividade: Você (ou dupla) deverá simular o funcionamento de um sistema operacional educacional por meio de scripts, relatórios e configurações em ambientes virtualizados. A atividade será dividida em etapas que abordam cada unidade temática da disciplina.

📌 Etapas da Atividade:

Etapa 1: Ambiente de Trabalho Virtualizado (Virtualização – 2ª unidade)

- Utilizar o VirtualBox, VMware ou Vagrant para configurar uma máquina virtual com Linux (Ubuntu ou Debian).
- Criar um relatório com os seguintes itens:
 - `cat /etc/os-release`: Analisar o arquivo que contem informações sobre o SO instalado.
 - `lscpu`: Analisar as informações detalhadas sobre a CPU do sistema.
 - `free -h`: Analisar o uso de memória RAM e swap do sistema.
 - `df -h`: Analisar o espaço em disco nas partições montadas.

Etapa 2: Introdução ao SO (3ª unidade)

- Criar relatório para identificar e descrever o tipo de arquitetura, estrutura de I/O e chamada do sistema
 - `uname -a`: Apresentar o nome do kernel, versão, arquitetura, nome do host;
 - `lsb_release -a`: Apresentar a distribuição e versão do sistema;
 - `file /sbin/init`: Verificar se o init é um executável, script ou link.
 - `Lsblk`: Lista os dispositivos de blocos (HDs, SSDs, partições).
 - `lspci | grep -i storage`: Identificar controladores de disco como SATA, NVMe, RAID.
 - `dmesg | grep -i usb`: Verificar se o sistema detectou pendrives, mouses, etc.
 - `strace -c ls`: Verificar quais chamadas o `ls` usou e quanto tempo cada uma levou.
 - `strace -e trace=open,read,write cat arquivo.txt`: Verificar quando o arquivo foi aberto, lido e escrito na tela.

Etapa 3: Gerenciamento de Processos (4ª unidade)

- Criar uma simulação em Python, C ou PowerShell de 2 algoritmos de escalonamento: FCFS e SJF.
 - Simule cenários para eles.
 - Incluir gráfico de Gantt simples, tempos de espera e turnaround.
 - Comparar resultados obtidos e concluir qual seria mais adequado para seu SO educacional.
 - `cat /proc/sched_debug | grep policy`: Verificar as políticas e prioridade em todos os processos.

Etapa 4: Threads e Concorrência (5ª unidade)

- Criar um pequeno programa com threads (Python ou C) para simular dois processos concorrentes (ex: produtor-consumidor).
- Explicar o modelo de threading utilizado (usuário ou kernel).
- Demonstrar sincronização por semáforo, mutex ou outro recurso.

Etapa 5: Gerenciamento de Memória (6ª unidade)

- Verificar o uso da memória:
 - `Free -m`: Analisar o uso da memória.
 - Explique as colunas (used | Buff | available)
 - `Vmstat 1 5`: Analisar estatísticas de memória, processos, CPU, etc.
 - Quantos processos esperando CPU?
 - `cat /proc/meminfo`: Verificar informações detalhadas da memória em tempo real.
 - `cat /proc/swaps`: Verificar os dispositivos de swap ativos e seu uso.
 - `swapon --show`: Verificar swap ativado com melhor formatação.

Etapa 6: Sistema de Arquivos (7ª unidade)

- Criar estrutura de diretórios com comandos `mkdir`, `touch`, `chmod`, `ls -l`.

```
# Criar diretórios
mkdir -p projeto/docs projeto/src projeto/bin

# Criar arquivos
touch projeto/README.txt
touch projeto/docs/manual.txt
touch projeto/src/app.sh

# Alterar permissões
chmod 755 projeto/src/app.sh      # Leitura/execução para todos, escrita só para o dono
chmod 644 projeto/docs/manual.txt # Leitura para todos, escrita só para o dono

# Listar com detalhes
ls -l projeto
ls -l projeto/docs
ls -l projeto/src
```

Entregáveis:

- Relatório técnico (PDF) com documentação das etapas, comandos, códigos-fonte, prints, gráficos e análises comparativas.
- Código-fonte dos simuladores (zip ou via repositório Git).
- Vídeo (opcional) de até 5 minutos demonstrando a execução dos scripts e VM.

Critérios de Avaliação:

| Critério | Peso |
|------------------------------------|------|
| Clareza e organização do relatório | 20% |
| Domínio dos conceitos teóricos | 20% |
| Execução técnica dos simuladores | 20% |
| Análise crítica dos resultados | 20% |
| Uso adequado de ferramentas | 20% |