

**UC Sistemas Operacionais
ICT/UNIFESP**

Prof. Bruno Kimura
bruno.kimura@unifesp.br
23/03/2023

TRAB_1: Implementação de um *Shell*

| | |
|-------------------------|--|
| Objetivo: | Implementar um interpretador de comandos em terminal para Linux. |
| Metodologia: | Trabalho em grupo de 4 (quatro) alunos(as) para desenvolvimento de protótipo em linguagem C. |
| Entregáveis: | <ul style="list-style-type: none">• Protótipo em arquivos .c• Vídeo-relatório (enviar link para o vídeo) |
| Data de entrega: | 13/04/2023 |
| Observação: | A autenticidade do trabalho será verificada. Cópias (entre grupos, de chatGPT e/ou de qualquer outra fonte da Internet) implicam em anulação. |

Requisitos do protótipo

O interpretador de comandos deverá:

1. Executar comandos unitários, com múltiplos parâmetros. Exemplo:
`# ls -la`
2. Executar comandos encadeados, utilizando o operador pipe “|” para combinar saída e entrada entre *n* comandos. Exemplo:
`# ls -la | grep "teste"`
`# cat /proc/cpuinfo | grep "model name" | wc -l`
3. Executar comandos condicionados com operadores OR (||) e AND (&&). Exemplo:
`# cat 12345 || echo "arquivo inexistente"`
`# ping -c1 www.unifesp.br.br && echo "SERVIDOR DISPONIVEL" || echo "SERVIDOR INDISPONIVEL"`
4. Executar comandos em background, liberando o shell para receber novos comandos do usuário.
`# ping -c5 www.unifesp.br &`
5. Redirecionar entrada e saída padrão de comandos para arquivos, utilizando “<” para entrada padrão e “>” para saída padrão. No caso do operador “>”, se o arquivo apontado para a saída já existir, o operador trunca a saída do comando sobre o conteúdo do arquivo, ou seja, sobrescreve o arquivo existente com o novo conteúdo. Exemplo:
`# sort < teste.txt > novo_teste.txt`
6. Redirecionar a saída padrão de comandos para arquivos, utilizando “>>”. Diferente do operador “>”, caso o arquivo de saída exista, o operador “>>” acrescenta a saída do comando ao fim do conteúdo do arquivo existente. Exemplo:
`# sort < teste.txt >> novo_teste.txt`
7. Combinar operadores. Exemplo:
`# sort < teste.txt | grep "x" > novo_teste.txt`
`# sort < teste.txt | grep "x" >> novo_teste.txt`

Vídeo-relatório:

- i. Apresentar explicação do trabalho desenvolvido: *o que, como e porque* foi feito da forma como foi.
- ii. Descrever de forma objetiva as contribuições de cada integrante do grupo no trabalho.
- iii. Apresentar uma auto-avaliação do grupo sobre o trabalho realizado.