

Atividade 07

Valor: 2,0 pontos na Segunda Nota

Primeira Chamada: 04/11/2025

Segunda Chamada: 11/11/2025

Questão 01 - 1,0 ponto - Deve ser feita em sala.

A) Gerenciamento de Processos em Background e Variáveis de Ambiente

Crie um script chamado `gerenciador_background.sh` que realize as seguintes tarefas:

1. **Leitura do Nome:** Solicite ao usuário que digite um nome para um novo processo (`read -p`).
2. **Criação do Processo:** Inicie um script auxiliar simples (que você criará no ponto 3) em *background* (`&`).
3. **Script Auxiliar (`contador.sh`):** Crie um script auxiliar chamado `contador.sh` . Este script deve:
 - Ser invocado no *background* pelo script principal.
 - A cada segundo, imprimir a mensagem: "Processo [Nome digitado] - Tempo: [Segundos]".
 - O laço de repetição (por exemplo, `while true`) deve ser encerrado através de um comando `kill` .
 - **Dica:** O script principal deve usar a variável `$_` para capturar o PID do `contador.sh` .
4. **Exibição e Espera:** O script principal deve:
 - Imprimir o **PID** do processo recém-criado usando a variável `$_` .
 - Usar o comando `sleep` por 10 segundos.
5. **Finalização:** Após os 10 segundos, o script principal deve usar o comando `kill` (com o PID armazenado na variável `$_`) para interromper o `contador.sh` .
6. **Confirmação:** Imprimir a mensagem: "Processo [Nome] finalizado (PID: [PID])."

Comandos-Chave a Utilizar: `read` , `&` , `sleep` , `echo` , `kill` , `$_`

B) Interface de Terminal e Controle de Cores

Crie um script chamado `status_interativo.sh` que demonstre o uso do `tput` para formatar a saída.

1. **Saída Limpa:** Limpe a tela do terminal no início do script.
2. **Variáveis:** Defina duas variáveis globais: `TITULO="Relatório de Status"` e `VERSAO="1.0"` .
3. **Cabeçalho Formatado:** Imprima o conteúdo da variável `TITULO` na **cor azul** (use `tput setaf 4`) e em **negrito** (`tput bold`).
4. **Posicionamento do Cursor:** Posicione o cursor na **linha 5**, coluna 10 (use `tput cup`).
5. **Cor e Versão:** Imprima a `VERSAO` em **vermelho** (`tput setaf 1`) e **sublinhado** (`tput smul`) na posição atual.
6. **Restauração:** Restaure as configurações de cor e estilo do terminal (use `tput sgr0`).
7. **Informação Dinâmica:** Use o comando `date` para obter a data e hora atual. Imprima esta data/hora na **linha 8**, coluna 10.
8. **Informação do Usuário:** Use o comando `id` (ou `whoami`) para obter o nome do usuário atual e imprima-o na **linha 10**, coluna 10.

Comandos-Chave a Utilizar: `tput clear`, `tput setaf`, `tput bold`, `tput smul`, `tput sgr0`, `tput cup`, `date`, `id` ou `whoami`.

Questão 02 - 1,0 ponto

Escreva um *script* chamado *alertaDiretorio.sh* que recebe como parâmetros um valor inteiro que serve como intervalo de tempo em segundos e um nome que indica o caminho de um diretório.

A cada intervalo, a quantidade de arquivos em um diretório será analisada. Caso a quantidade de arquivos se altere entre duas verificações, o *script* deve atualizar um arquivo chamado *dirSensors.log* com as seguintes informações:

1. A data que a alteração foi percebida.
2. Quantos arquivos existiam.
3. Quantos existem agora.
4. Quais foram alterados, adicionados ou removidos.

Na mesma pasta da atividade, crie um diretório chamado *diretorioMonitorado* para testar. Exemplo de execução:

```
$ ./alertaDiretorio.sh 5 diretorioMonitorado
[25-01-2021 12:59:51] Alteração! 3->2. Removidos: notas.txt
[25-01-2021 13:04:51] Alteração! 2->4. Adicionados: a.txt, b.txt
[25-01-2021 13:09:51] Alteração! 4->3. Removidos: a.txt
[25-01-2021 13:14:51] Alteração! 3->2. Removidos: b.txt
```

Na correção, usando *screen* ou *tmux*, crie duas telas divididas na vertical. Na tela 01, você deixará o script executando. Na tela 02, você estará no *diretorioMonitorado*. Ao criar ou remover arquivos na tela 02, as mensagens correspondentes devem aparecer na tela 01.