

Nome: Gabriel

Instruções

- A prova é individual e sem consulta.
- Não é permitido o uso de qualquer dispositivo eletrônico.
- A interpretação do enunciado é parte da resposta.
- As respostas devem ser escritas a caneta azul ou preta.

1. O comando `make` é utilizado para: (5 pontos)
 - (a) Gerar arquivos de configuração e testes
 - (b) Atualizar repositórios remotos
 - ☒ (c) Automatizar tarefas baseadas em dependências
 - (d) Executar scripts shell
2. Qual comando Git é usado para enviar mudanças para um repositório remoto? (5 pontos)
 - ☒ (a) `git push`
 - (b) `git commit`
 - (c) `git clone`
 - (d) `git merge`
3. O que faz o seguinte comando no Bash?:
`chmod +x script.sh` (5 pontos)
 - (a) Altera a permissão de leitura do arquivo
 - (b) Compila o arquivo `script.sh`
 - ☒ (c) Torna o arquivo executável
 - (d) Cria um link simbólico para o script
4. Assinale (V) para verdadeiro ou (F) para falso. (10 pontos)
 - (F) O comando `cd ..` move o usuário para o diretório raiz.
 - (V) O comando `git status` exibe arquivos modificados e não “comitados”.
 - (F) O Git é um sistema de controle de versões centralizado.
 - (V) Em Bash, `&&` permite encadear comandos que devem ser executados sequencialmente, mas apenas se o anterior for bem-sucedido.
 - (V) O Git permite o trabalho *offline*, pois os *commits* são armazenados localmente.
 - (V) O operador `|` em Bash é utilizado para redirecionar a saída de um comando para a entrada de outro.
 - (V) Branches em Git são úteis para desenvolver novas funcionalidades sem afetar o código principal.
 - (V) No Bash, a variável `$0` representa o nome do script que está sendo executado.
 - (V) O comando `git pull` é usado para baixar e aplicar mudanças de um repositório remoto.
 - (F) O comando `touch arquivo.txt` é usado para mover arquivos de um diretório para outro.
5. Explique o que é um repositório Git e qual a importância do comando `git commit`. (5 pontos)

Um repositório Git é um local de desenvolvimento usado para controle de versões e trabalho em equipe. Git commit é responsável por consolidar uma nova versão local com uma mensagem associada. Posteriormente, as mudanças podem ser enviadas com `git push`.

6. Relacione os comandos ao seu comportamento correto:

(10 pontos)

- (a) `git status`
- (b) `git log`
- (c) `git merge`
- (d) `git checkout`
- (e) `pwd`
- (f) `echo`
- (g) `mv`
- (h) `cat`
- (i) `grep`
- (j) `cut`

- (C) Une dois ramos de desenvolvimento
- (B) Exibe o histórico de commits
- (I) Busca por padrões de texto em arquivos
- (F) Envia mensagens para a saída padrão
- (G) Move ou renomeia arquivos e diretórios
- (H) Exibe o conteúdo de um arquivo
- (A) Exibe informações sobre alterações e arquivos modificados
- (D) Alterna entre branches ou restaura arquivos
- (E) Exibe o diretório atual
- (J) Exibe partes/seções de cada linha de um arquivo

7. Você executou um script sem querer, e ele colocou algumas imagens no diretório `work`, que fica dentro do diretório atual. Escreva um script/comando que remova todos os arquivos dentro de `work/` cujos nomes iniciem com `out_0` ou `out_1` e que terminem com `.png`.

(15 pontos)

```
rm work/out_0,1/*.*.png
```

8. O script `weather.sh` no diretório atual exibe a previsão do tempo na sua região. Quando executado, o script mostra a previsão de 4 dias, no formato “dia: previsão”, onde cada dia está em uma linha. Escreva um script/comando que mostre quantos dias possuem previsão para “Temporal”.

(15 pontos)

```
./weather.sh | grep "Temporal" | wc -l
```

9. O que é um Makefile? Dê um exemplo básico de uma regra nele e explique seu funcionamento.

(5 pontos)

É um arquivo com regras para automatizar ações considerando suas dependências.

Sempre que a dependência "primo.cpp" for alterada, o alvo "primo" será reconstruído usando a ação "g++..."

```
primo: primo.cpp
g++ primo.cpp -o primo
```

10. Descreva a diferença entre os operadores Bash `>` e `>>` com um exemplo prático de uso.

(10 pontos)

Ambed redirecionam a saída de um comando para um arquivo. `>` sobrescreve o arquivo com a nova saída e `>>` adiciona a saída ao fim do arquivo.

```
# cat fruta
banana
# echo abacate > fruta
# cat fruta
abacate
```

11. Escreva um script shell para compilar todos os seus programas C++, dentro do diretório `~/ifpr-bcc/1sem/prog/`.

(15 pontos)

```
#!/bin/bash
pushd ~/ifpr-bcc/1sem/prog/
for arquivo in *.cpp; do
    g++ $arquivo -o ${arquivo%.cpp}.out
done
popd
```