

Estatísticas de utilizadores em bash

Licenciatura em Engenharia Informática

Sistemas Operativos

Docente: Nuno Lau

Isadora Loredo, 91322

Gonçalo Matos, 92972

Ano letivo 2019/2020

Índice

[**Introdução**](#_rv9w9k5nnh54) **3**

[**Soluções implementadas**](#_1tavmuw26ztx) **4**

[**userstats.sh**](#_7bi8dl2ah90) **4**

[**countUsers()**](#_db5qkcljfw8n) **4**

[**detailSessions()**](#_2gf3v0bqoms6) **4**

[**dateConversion()**](#_eg688crz224m) **4**

# Introdução

Neste projeto foi-nos proposto que desenvolvêssemos dois *scripts* em bash para a recolha de algumas estatísticas sobre o modo como os utilizadores utilizam o sistema operativo, nomeadamente o número de sessões, os tempos que estas duraram e algumas relações entre estes e outras variáveis como o grupo a que pertencem, que serão explicadas em detalhe neste relatório.

Apresentados nesta ordem, o primeiro é responsável pela consulta e processamento dos dados, transformando-os em informação que é impressa no terminal e pode ser escrita num ficheiro, que posteriormente pode ser processado pelo segundo, que tem como objetivo comparar dois ficheiros resultantes da execução do primeiro e mostrar ao utilizador a relação entre a informação presente em ambos.

Todo o código foi desenvolvido de raiz, tendo por base os conhecimentos adquiridos nas aulas práticas e a pesquisa realizada durante o decorrer do projeto, e como objetivo a correspondência dos resultados com o *output* esperado, descrito no guião do projeto, com vários exemplos de execução.

# Soluções implementadas

## userstats.sh

A execução deste programa começa por inicializar algumas variáveis, cujo propósito será descrito mais à frente aquando da sua utilização, seguido do processamento das opções de execução introduzidas pelo utilizador.

Para as opções que aceitam argumentos (g, u, s e f), é lido o argumento (elemento seguinte do array que guarda os argumentos - *${args[a + 1]}*) e para as restantes opções que são de filtragem (r, n, t, a e i) é verificado se não estão a ser adicionadas opções incompatíveis (apenas pode existir uma opção n, t, a ou i, que podem ser conjugadas (opcionalmente) com a r e caso isto não se verifique, são adicionadas à variável de filtragem.

De seguida são invocadas duas funções, uma que cria uma listagem dos utilizadores que vão ser processados (tendo em conta os utilizadores que constam nos dados disponíveis para consulta e os filtros aplicados) - *countUsers()* - e outra para processar os detalhes associados às sessões de cada utilizador listado anteriormente - *detailSessions()*.

### countUsers()

Esta função começa por verificar se é necessário haver filtragem dos utilizadores por grupo, através do acesso ao ficheiro */etc/group* do sistema, que é filtrado em busca do nome do grupo passado como argumento, obtendo a lista de utilizadores pertencentes ao mesmo. Caso não haja filtragem por grupo, os utilizadores a serem processados serão todos para os quais existam registos, sendo estes dados obtidos através da consulta da primeira coluna do *output* do comando *last* (que corresponde ao número de utilizadores), removendo os repetidos e ainda alguns utilizadores que estão associados a processos do sistema e que não achamos pertinente o seu *output* (nomeadamente o *shutdown*, *reboot*, *root* e *\_mbsetupuser*). Em ambos os cenários descritos, a lista dos utilizadores é armazenada na variável *users*.

Por fim, caso seja necessária a filtração por utilizador, o *array* criado no processo anterior será percorrido e os elementos que não correspondem à expressão *regex* passada como argumento serão eliminados do mesmo.

Finalmente está concluída a função e a lista de utilizadores cujos dados das sessões serão analisados está compilada.

### detailSessions()

#### dateConversion()