Universidad de Costa Rica
Escuela de Ciencias de la Computación e Informática
CI-0124 Computabilidad y Complejidad, grupo 01
Fecha: día/mes/año, I ciclo lectivo 2019

Tarea de Programación # 2: práctica de AWK

En esta tarea de programación vamos a utilizar scripts de AWK para resolver dos problemas distintos, pero que nos darán un acercamiento a las capacidades y poder computacional de AWK. Esta tarea se puede resolver en grupos de 2 personas.

1. Problema # 1 (30 %)

El archivo `censo2000.txt` contiene datos extraídos de un censo que se llevó a cabo en Estados Unidos en 2000. Los datos mostrados corresponden a condados del estado de Washington. Descargue este archivo y examínelo para que se de una idea de su contenido. El archivo consta de 4 columnas de datos separadas por marcas de tabulador que corresponden al nombre del condado, población, área acuática y área terrestre, ambas en millas cuadradas. La primera línea del archivo es el encabezado de las columnas.

Escriba un script AWK llamado `estadisticasCondados.awk` que procese el archivo y para cada condado imprima su nombre, la densidad de población por milla cuadrada terrestre, y el porcentaje de la superficie del condado que es agua. Para esto último, no olvide que el porcentaje debe calcularse sobre la totalidad de la superficie del condado (agua + tierra).

Al final, luego de haber impreso los anteriores datos para cada condado, su script debe imprimir lo siguiente:
Nombre del condado y el valor de la mayor densidad de población.
Nombre del condado y el valor de la menor densidad de población.
Nombre del condado y el valor del mayor porcentaje de agua.
Nombre del condado y el valor del menor porcentaje de agua.

Por ejemplo, al final debe salir algo como:
Mayor densidad de población: King County con 817 hab/millaˆ2
Menor densidad de población: Columbia County con 4.67 hab/millaˆ2
... etc.

2. Problema # 2

El cluster arenal.ecci.ucr.ac.cr de la ECCI ha estado sujeto a ataques del exterior con el fin de tratar de averiguar el password de algún usuario administrativo, como el root, probablemente para hackearlo. El comando del root `[root@arenal ]# lastb -ia` da un listado de los intentos fallidos de login, los cuales se van almacenando en el archivo del sistema `/var/log/btmp`.

Este listado de intentos fallidos fue extraído del sistema y guardado en `arenalBadLogin.txt`. Observe que el archivo contiene campos separados por uno o varios espacios en blanco. Coloque una expresión adecuada en la variable interna FS de AWK para acceder con facilidad a los campos del archivo. Usted no puede alterar (editar) el archivo de entrada `arenalBadLogin.txt`.

2.1. Parte a) (20 %)

Escriba un script de AWK para procesar este archivo y generar un reporte que cuente el número de veces que cada dirección IP ha intentado hacer un login.

2.2. Parte b) (20 %)

Mejore la calidad de la salida anterior por medio de ordenar de mayor a menor las direcciones IP con base en el número de intentos fallidos (no se trata de ordenar por dirección IP, sino por el número de intentos). En el listado debe verse cada dirección IP acompañada del número de intentos, empezando por la IP con más intentos. Para lograr esto, estudie las funciones asort y asorti de AWK. Puede ver https://www.gnu.org/software/gawk/manual/html_node/Array-Sorting-Functions.html. También es posible que necesite crear un nuevo arreglo con subíndices apropiados y luego extraer subhileras (ver https://www.gnu.org/software/gawk/manual/html_node/String-Functions.html) para un resumen de las funciones principales que se aplican a las hileras.

2.3. Parte c) (30 %)

Ahora se debe agregar información del sitio IP tal como país, ciudad, nombre de la red. Para ello, experimente por separado con el comando whois de Linux. Lea el manual en línea o una guía de su uso. Además, considere la siguiente ayuda.

En AWK se pueden ejecutar comandos del sistema operativo de dos maneras: mediante la llamada `system()`, o mediante la construcción de una hilera que contiene el comando a ejecutar. Además, la salida del comando en la hilera se puede pasar por un pipe a la función getline que la captura y la guarda en una variable. Posteriormente, esa variable se puede utilizar dentro de AWK para cualquier propósito, como imprimirla junto con otros datos relacionados. Investigue todo esto (ver https://www.gnu.org/software/gawk/manual/html_node/Getline_002fPipe.html) y utilícelo para crear un listado similar al siguiente.

```
116.234.39.28  Failed logins=  13
netname:        CHINANET-SH
descr:          CHINANET Shanghai province network
descr:          China Telecom
descr:          No.31,jingrong street
descr:          Beijing 100032
country:        CN
address:        No.31 ,jingrong street,beijing
address:        100032
address:        Room 2405,357 Songlin Road,Shanghai 200122
country:        CN

37.204.212.15  Failed logins=  12
netname:        NCN-BBCUST
descr:          NCNET Broadband customers
country:        RU
address:        National Cable Networks
address:        Nagatinskaya str., 1, bldn. 26
address:        117105 Moscow, Russia
org:            ORG-NCN1-RIPE
descr:          NCNET
```

```
186.4.13.104  Failed logins=  10
address:     350 mts Oeste de Pops Sabana, 200 mts Sur frente a entrada de UCIMED, 0, 0
address:     0 - San Jose - SJ
country:     CR
address:     San Jose, 1234,
address:     10108 - San Jose - San Jose
country:     CR
address:     350 mts Oeste del Ministerio de Agricultura, frente a entrada de UCIMED., , Saba
address:     10108 - San Jose - SJ
country:     CR
```

Como el listado completo puede ser largo y tardar un rato, basta con que imprima la información de las primeras 10 IP con mayor número de logins fallidos. Combine la salida de whois con grep para seleccionar aquellas líneas que puedan identificar al país o ciudad. Algunas veces la salida de whois produce atributos que inician con mayúscula o minúscula (ej. Country, country). Investigue alguna forma de decirle a grep que ignore estas diferencias.

Una expresión regular de tipo OR en grep debe ser protegida para que el símbolo de pipe (|) no se confunda con el de OR. Investigue cómo se puede hacer esto y utilícelo al construir su expresión regular.

Entregar en Moodle el miércoles 24 de abril. Suba un archivo comprimido con los documentos en LaTeX, imágenes y el pdf respectivo. Incluya en el archivo comprimido los scripts fuente (.awk) de cada caso.