FACULTAD
DE INGENIERIA
Universidad de Buenos Aires

75.08 Sistemas Operativos 2do. Cuatrimestre 2013 **PARCIAL Practico** **29-10-2013**

Colocar Nombre y Padrón en cada hoja de enunciado que se entrega. Si no va a resolver un ejercicio, entrega el enunciado con los datos completos y la leyenda: "NO RESUELVO".

Colocar Nombre y Padrón en cada hoja de resolución. Resolver cada ejercicio en Hoja APARTE

Padrón	93155	**Apellido y Nombre**	BUCHWALD MARTÍN	**Nota**	A

Perl

Tema 2

Restricciones: Resolver usando PERL. **NO USAR** comandos Unix, DOS, o de cualquier otro sistema operativo bajo ningún concepto; ello limita la portabilidad del script perl. **NO USAR** expresiones regulares para separar o juntar los campos de un registro, para ello utilizar funciones de librería Perl adecuadas para la separación o corte de campos y caracteres. **NO USAR** archivos auxiliares.

Archivos: Todos los archivos con separador de campos punto y coma y separador de registros new line.

- Un archivo de saldos iniciales **(SALDINIC)**: *Nombre de la cuenta; Nro. de Cuenta; Saldo Inicial.*
Dos Ejemplos: ***Caja;1010;1234,56*** *Banco Rio cuenta particular;1020;78901,23*

- N archivos con movimientos de cuenta (*.**mov**): *Nro. de Cuenta; Fecha del Movimiento; Débito; Crédito.*
Cinco Ejemplos: ***1020; 27-10-13;1.23;0*** *1020;27-10-13;0;100* ***1010;28-10-13;34.56;0*** *1010; 28-10-13;0;1300* ***1010;28-10-13;0;80.08***

Asumir que el archivo saldinic y los de movimientos vienen sin errores de datos ni errores de formato.
Asumir que no vienen movimientos de cuentas inexistentes (todos los movimientos son de cuentas que están en saldinic)
Abrir y cerrar adecuadamente TODOS los archivos. Si el archivo de salida existe, reemplazarlo. Si no existe, crearlo.

Invocación El programa se invoca con dos parámetros: el **Parámetro 1:** directorio donde se encuentra el archivo saldinic y en el **Parámetro 2**: nombre del archivo de salida.

Proceso: **Abrir y Leer** TODOS los archivos *.mov que se encuentran en el directorio corriente. **Acumular** usando hash (emplear como clave el nro. de cuenta) los montos de debito y crédito. Los débitos se restan, los créditos se suman.
Cuando se terminan todos los archivos, **mostrar** por estándar output UNA línea por cada clave-valor del hash.
Luego **preguntar** al usuario por estándar input si desea calcular el nuevo saldo de cada cuenta. Si contesta que NO, terminar el programa.
Si contesta que SI **validar usando file test** que el archivo **saldinic** exista en el directorio pasado como parámetro 1 y **validar** que se haya pasado algún valor en el parámetro 2, el cual se usará como nombre de archivo de salida.
Si hay algún error, **mostrar** estándar output un mensaje descriptivo del error y terminar el programa.

Final: **Grabar** los nuevos saldos en el archivo de salida con: *Nombre de la Cuenta; Nro. de Cuenta; Nuevo Saldo* (1 registro para cada cuenta)
Nuevo Saldo cuenta X = Saldo Inicial de la cuenta X + valor del hash para la clave X o cero si no existe en el hash.
Dos Ejemplos: ***Caja; 1010;2580.08*** *Banco Rio cuenta particular; 1020;79000*

A–

MARTÍN
BUCHWALD

PADRÓN : 93155

Perl

falta la 1º línea del programa Perl

¿de dónde salió este vector?

SOLUCION:
@archivos = <*.mov>;
foreach $archivo (@archivos)
}

```
%acumulado;
 foreach $archivo (@path) {
  if (! $archivo ~= /.*\.mov$/)){
     next;
  }
   open (ENTRADA, "<$archivo");
   while ($registro = <ENTRADA>) {
    #obtengo los índices donde se encuentran los separadores
    $primer_separador = index($registro, ";", 0);
    $segundo_separador = index ($registro, ";", $primer_separador +1);
    $tercer_separador = index ($registro, ";", $segundo_separador+1);
   #obtengo los valores de los campos que me interesan
   $cuenta = substr ($registro, 0, $primer_separador);
   $debito  = substr ($registro, $segundo_separador+1, $tercer_separador
                      - $segundo_separador -1);
  $credito = substr ($registro,$tercer_separador+1);
   $suma = $debito - $credito;
   if ( exists ($acumulados{$cuenta})) {
       $acumulados{$cuenta} += suma;
  }else{
       $acumulados{$cuenta} = suma;
  }
 close(ENTRADA);
}
```

Todo esto se soluciona en 1 sólo línea
@campos = split (/;/, $registro) éste
separador
divide
lo pone en un vector

```
print "Modificaciones:\n"
Foreach $cuenta (Keys(%acumulado)){
    print $cuenta.":".$acumulado{$cuenta}."\n";
}

print "Desea guardar las modificaciones?\n"
$respuesta=<INPUT>;    STDIN
                       chomp($respuesta)
if ($respuesta eq "No" ){
    exit 0;
}

if ( $#ARGV +1 != 2) {
    print "cantidad de parametros invalido \n";
    exit 1;
}

($directorio, $ruta_salida) = @ARGV;
$ruta_saldinic = $directorio."/saldinic";
if (! -f $ruta_saldinic ){
    print "No existe el archivo de saldos iniciales en el directorio ingresado\n";
    exit 1;
}
# Una vez realizadas las validaciones, guardamos las modificaciones
open (SALDINIC, "<$ruta_saldinic");
open (SALIDA, "$ >$ruta_salida");

# sigo en otra hoja
```

MARTÍN BUCHWALD	PADRÓN: 93155	

```
While ($registro = <SALDINIC>){
   $primer_separador = index($registro, ";", 0);
   $segundo_separador = index ($registro, ";", $primer_separador +1);

   $nombre = substr($registro, 0, $primer_separador);
   $cuenta = substr($registro, $primer_separador +1, $segundo_separador -
                                          $primer_separador -1);
  $saldo = substr ($registro, $segundo_separador +1);
  if ( exists ($acumulado{$cuenta})) {
    $saldo += $acumulado{$cuenta};
  {
   print SALIDA $nombre.";".$cuenta.";".$saldo."\n";
}
 close (SALIDA);
 close(SALDINIC);
```

idem ant

SPLIT