

Solución del examen de Agosto de 2009

Ejercicio 1

a)

Logearse por ssh con su usuario correspondiente al servidor “introcomp-server.fisica.edu.uy”.

```
> ssh usuario@introcomp-server.fisica.edu.uy (pide luego la clave)
```

Crear la carpeta examen-agosto-<usuario> y cambiar su directorio de trabajo a esa carpeta.

```
> mkdir examen-agosto-usuario
```

```
> cd examen-agosto-usuario
```

Listar desde su ubicación actual los permisos de esa carpeta (independiente de donde se encuentre ella).

```
> ls -l ..
```

b)

Listar todos los archivos de la carpeta /etc que comiencen con a.

```
> ls /etc | grep ^a
```

Idem la parte anterior pero todos los que comiencen con b o c.

```
> ls /etc | grep ^b ; ls /etc | grep ^c
```

c)

Listar el contenido de las carpetas “fortran” y “c” dentro de /home/introcomp-files/ de forma que cada listado quede guardado en un archivo “lsfortran.txt” y “lsc.txt” respectivamente.

```
> ls /home/introcomp-files/fortran > lsfortran.txt
```

```
> ls /home/introcomp-files/c > lsc.txt
```

Cambiar todos los permisos necesarios para que otros puedan escribir en el archivo lsfortran.txt .

```
> chmod o+w lsfortran.txt , también se debe tener cuidado de cambiar el permiso de la jerarquía ascendente de las carpetas en el sistema de archivos.
```

d)

Concatenar los dos archivos creados en la parte c en uno solo llamado “programas.txt”

```
> cat lsfortran.txt lsc.txt > programas.txt
```

En una línea , contar la cantidad de líneas del archivo programas.txt que identifican a los programas en C (los que terminan en .cpp).

```
> cat programas.txt | grep .cpp$ | wc -l
```

e)

Escribir en un línea lo necesario para desplegar la sig. salida independientemente de que usuario lo ejecute: Yo soy el usuario <usuario>

```
> echo "Yo soy el usuario $USER"
```

Escribir en una línea lo necesario para que la última fila del archivo programas sea: usuario: <usuario> , fecha: <fecha actual>

```
> echo "usuario: $USER , fecha: `date`"
```

f)

Escribir un script de Linux que guarde el contenido del archivo programas.txt en otro archivo programas-reversa.txt pero con el contenido en un orden invertido.

(Sug. Filtrar el contenido de programas.txt de forma de ir recorriendo sus líneas de la última hasta la primera de la siguiente forma: Para la línea i (a partir de la última), quedarse con las últimas i líneas y luego con la primera de ese resultado)

```
#!/bin/bash
```

```
# guardo en nr_lineas la cantidad de lineas del archivo resultado de wc
```

```
nr_lineas=`wc -l programas.txt|cut -d ' ' -f 1`
```

```
# para cada nro i de linea, obtengo las ultimas i lineas, luego la primera del resultado y
```

```
# redirecciono el resultado al archivo programas-reversa.txt
```

```
for((i=1; i<=$nr_lineas; i++))
```

```
do
```

```
  tail -$i programas.txt | head -1 >> programas_reversa.txt
```

```
done
```

```
#eof
```

Ejercicio 2

program snell

```
integer :: N, planeOfrefraction    ! N -> nro de subdivisiones del medio refractivo
                                     ! planeOfRefraction -> indice que recorre los planos donde evaluo la refraccion
real :: L                          ! ancho del medio refractivo
real :: angleOfIncidence,angleOfRefraction    ! angleOfIncidence -> angulo de incidencia del rayo sobre la cara izq de la
                                               ! interfaz
                                               ! angleOfRefraction -> angulo del rayo refractado sobre la cara drcha
                                               !de la interfaz

real :: lastAngleOfRefraction
real :: error, deltaAngle, h, n1, n2    ! deltaAngle -> diferencia entre el angulo del rayo de salida en x=L para un N y N/2
```

```
! ----- Instancio las vars -----
```

```
L = 1.0
```

```
N = 2
```

```
angleOfIncidence = 1.5
```

```
lastAngleOfRefraction = 0
```

```
deltaAngle = 3.14
```

```
error = 0.1
```

```
! Comenzando por N=2, itero sobre las subdivisiones del plano para hallar el angulo en x=L
```

```
! Repito el proceso para N=2N, N=4N hasta que no haya una diferencia entre el angulo de salida sucesivos mayor a error
```

```
do while (deltaAngle > error)
```

```
  h = L / N
```

```
  x = 0
```

```
  n1 = 1
```

```
  angleOfIncidence = 1.5
```

```
  do planeOfRefraction = 1,N+1
```

```
    n2 = indexOfrefraction(x)
```

```
    angleOfRefraction = asin(sin(angleOfIncidence)*n1/n2)
```

```
    x = x + h
```

```
    n1 = n2
```

```
    angleOfIncidence = angleOfRefraction
```

```
    print *, "angle refr: ",angleOfRefraction," indice: ",n2
```

```
  end do
```

```
  N = 2*N
```

```
  deltaAngle = abs(angleOfRefraction - lastAngleOfRefraction)
```

```
  lastAngleOfRefraction = angleOfRefraction
```

```
  print *, "angulo de refr salida: ",angleOfRefraction," delta: ",deltaAngle
```

```
end do
```

```
!-----
```

```
CONTAINS
```

```
real function indexOfRefraction(x)
```

```
  real, intent(in) :: x
```

```
  real, parameter :: indexIni = 1.2 , indexEnd = 1.9
```

```
  indexOfRefraction = indexIni + (indexEnd - IndexIni)*x/L
```

```
end function indexOfRefraction
```

```
end program snell
```

