Trabajo práctico de Laboratorio 02

Introducción al shell del sistema GNU/LINUX

Introducción a la computación
Departamento de Ingeniería de Computadoras
Facultad de Informática - Universidad Nacional del Comahue

Lectura obligatoria:

- Apunte del shell de Linux: `http://pedco.uncoma.edu.ar/mod/resource/view.php?id=207175`
- Apunte introductorio a **BASH**: `http://pedco.uncoma.edu.ar/mod/resource/view.php?id=244968`
- Linux Man Pages Online: `https://linux.die.net/man/`

1. Administración de procesos

1. Cree un archivo con nombre `dance.sh` con el siguiente contenido. Verifique que ha copiado correctamente cada linea.

```
#!/bin/sh

while true; do
    printf "\t(>'-')>\r"
    sleep 0.5
    printf "\t<('-'<)\r"
    sleep 0.5
done
```

2. Agregue permisos de ejecución utilizando el comando `chmod +x dance.sh`.
3. Ejecute el programa ingresando en la terminal `./dance.sh`.
4. Abra otra ventana y utilizando el comando `ps a` encuentre el `PID` del proceso del programa `dance.sh`. Termine el programa utilizando el comando `kill` *`PID_DEL_PROCESO`*.
5. Ejecute el comando `file` sobre el archivo `dance.sh` y `/bin/ls` ¿Cuál es la función del comando `file`?

2. Lenguajes compilados e interpretados

2.1. Lenguajes interpretados

6. Cree un archivo con nombre `calcular.sh` con el siguiente contenido. Verifique que ha copiado correctamente cada linea.

```
#!/bin/sh
max=${1:-10}
num=0
for i in $(seq 1 "$max");do
    num=$((num + 3))
done
echo "El resultado de sumar 3 unas $max veces es: $num"
```

7. Agregue permisos de ejecución utilizando el comando `chmod +x calcular.sh`. Mida el tiempo de ejecución para 100, 100 000 y 1 000 000 utilizando el comando `time ./calcular.sh $NUM` (remplace `$NUM` por cada uno delos valores).

2.2. Lenguajes compilados

8. Cree un archivo con nombre `calcular.c` con el siguiente contenido. Verifique que ha copiado correctamente cada linea.

```
#include<stdio.h>
#include<stdlib.h>
#include<stdint.h>

int main(int carg, char **varg)
{
    uint64_t max = 10, i, num = 0;
    if (carg > 1) {
        max = atoi(varg[1]);
    }
    for (i = 0; i < max; i++) {
        num = num + 3;
    }
    printf("El resultado de sumar 3 unas %d veces es: %d\n", max, num);
    return 0;
}
```

9. Cree un archivo ensamblador optimizado con el comando `gcc calcular.c -S -O3` (`-O` es la letra "O" mayuscula). Inspeccione el contenido del archivo `calcular.s`.

10. Cree un archivo ejecutable con el comando `gcc calcular.s -o calcular` (`-o` es la letra "o" minuscula). Ejecute el comando `file calcular` ¿Qué información retorna el comando?

11. Mida el tiempo de ejecución para 100, 100 000 y 1 000 000 utilizando el comando `time ./calcular $NUM` (remplace `$NUM` por cada uno delos valores).

2.3. Comparando resultados

12. ¿Cuál de los dos ejecutables fue más rápido?

13. Modifique ambos programas para que sumen el número 42 en lugar de 3. Verifique que el cambio se realizo con exito sumando 100 veces.

14. ¿Cuál considera que fue más simple de modificar?

15. Indique en sus palabras cuales son las ventajas y desventajas de los lenguajes compilados y los interpretados.