

Especificaciones Técnicas

Compilador

gcc (Fedora 21) 4.9.2

Copyright (C) 2015 Free Software Foundation, Inc.

Computadora

HP NoteBook 14

RAM: 4 GB

Disco Duro: 320 GB

Procesador: Intel® Core™ i5-4210U

Ficheros

- comandos.go
- struct.go
- main.go

Estructuras

Utilizadas

```
type
propiedadesDisco
struct {
    Size int
    Path string
    Name string
    Unit string
}
type struct
{
type mbr struct {
    Size      int64
    Time      [25]byte
    Disk_signature int8
    Partitions [4]partition
}
```

```
type partition struct {
    Status byte
    Type   byte
    Fit    byte
    Start  int64
    Size   int64
    Name   [16]byte
}
type montados struct {
    Size           int
    Path           string
    Name           string
    Identifier      string
    Created        string
    particiones_montadas int
}
```

```
typedef struct
```

```
{
int id3;
char
data3[12];
int stado3;
}bloco_p3;
```

```
type propiedadesPart struct {
```

```
    Size  int64
    Unit  byte
    Path  string
    Type  byte
    Fit   [1]byte
    Delete bool
    Name  string
    Add   bool
}
```

```
typedef struct
```

```
{
    int id;
    int
    indice1;
```

Variables globales

```
var comandoos string = ""
```

MÉTODOS USADOS

```
func fileexist()
```

- Verifica si existe el archivo que se creará

```
func firstone()
```

- primer ajuste devuelve el no. de bloque donde se escribira el archivo

```
func worstone()
```

- peor ajuste devuelve el no. de bloque donde se escribira el archivo

```
func bestone()
```

- mejor ajuste devuelve el no. de bloque donde se escribira el archivo

```
Func savefile()
```

- _ Almacena el archivo

```
func deletefile()
```

- Elimina el archivo

```
func searchfile()
```

- Busca el archivo y devuelve su información

```
func main()
```

- funcion primaria, que abre el archivo de entrada, lo lee y llama metod de mostrar, y readfile

- Reportes de ca disco

```
Void mrep1()
```

- Reportes de cada partición

```
Void archsp1()
```

```
func mostrar(i string)
```

- separa las lineas leidas y manda a llamar a funcion entro_todo

```
func obtener(txt string) string
```

-splitea la linea independientemente si se trabajo en Windows o Linux

func entro_todo(linea_cm []string)

-lee el primer registro del arreglo de línea leida y lo manda a distintas funciones para su analisis

func readFile(nombre string)

-separa el archivo de entrada en líneas separadas para su analisis

func separar(txt string)

-metodo que separa por espacios

func IDmount(path string) string

-Identifica cada estructura de mount

func disco_mount(path string, name string)

-llena la estructura del mount

func verificaMount(path string, name string)

-verifica la existencia dle mount

func comando_mounts(com []string)

-analiza las propiedades leidas del archivo de entrada y llama a la función disc mount

func getPath(p string)

obtiene la ruta enviada

func compareBytes(str1 string, str2 string) bool

-compara cadenas de strings en bytes

func imp_mounts()

-imprime la informacion del mount

func comando_fdisks(com []string)

-analiza las propiedades leidas del archivo de entrada y llama a la función dsik

func BytesToString(b []byte) string

-separa un arreglo de strings en bytes

func obtenerNombrDisc(p string) montados

-obtiene los nombres de las rutas analizadas

func createPartition(r *mbr, p propiedadesPart, ds montados)

-crea las particiones en binario

func calcPart(parti [4]partition) (int, int, int)

-calcula el espacio de cada particion

func ArchivoLeido(nombre string) mbr

func consolaDisco(m mbr)

-analiza la línea de entrada con respecto a los discos