

PrioridadMayorContinuaReubicable .sh(1)

Table of Contents

ALUMNO	2
INFORMACIÓN	3
FUNCTIONS	3
DETAILS	4
Script Body	4
DEV_ImprimirColores	7
DEV_modificarMemoria	7
abrirInforme	8
ajustarMemoriaParaElProceso	9
anadirCola	10
aniadirProcesoAMemoria	10
aniadirSiguienteProcesoACPU	11
asignarColorProceso	12
asignarColoresTabla	12
breakpoint	13
calcularPosProceso	13
calcularPosTodosProcesos	14
cargaDatos	14
colorearBarraTiempo	16
comprobacionDirectorio	16
comprobarDatosFichero	16
comprobarRango	17
comprobarRepite	18
comprobarSN	18
comprobarSiElProcesoEnCPUHaTerminado	18
convertirFicheroColorEnBlancoNegro	19
creacionDeDirectorios	19
datosAleatorios	20
datosFichero	20
datosManualProcesos	22
debug	23
deleteGeneratedFiles	23
dibujarEstadoCPU	23
dibujarMemoria	24

ejecucion	24
ejecutarUnCiloDeCPU	26
eliminarCola	26
eliminarPosProceso	26
eliminarProcesoDeMemoria	27
encontrarHuecoEnMemoria	27
escribeDatos	27
establecerPrioridad	28
finMain	28
fin_programa	29
forzarCierre	29
global	30
imprimirAviso	33
imprimirCabecera	33
imprimirErrorCritico	34
imprimirLCyan	34
imprimirTabla	35
imprimirTablaPredeterminada	35
inicializarArrays	36
main	36
medirTiempo	39
menuAlgoritmo	39
nularColumna	39
numAleatorio	40
ordenarProcesos	40
renombrarDatosEntrada	40
reubicarProcesos	41
salirPorErrorCritico	41
scanfNum	42
scanfNumMinMax	43
scanfSiNo	43
scanfString	44
selectorFichero	44
truncarBarraCPU	45
truncarMemoria	46
vaciarMemoria	46

ALUMNO

Jorge Ruiz Gómez

INFORMACIÓN

Script que simula el comportamiento de un algoritmo

Prioridad Mayor - Memoria Continua Reubicable

FUNCTIONS

```
DEV_ImprimirColores
DEV_modificarMemoria
abrirInforme
ajustarMemoriaParaElProceso
anadirCola
aniadirProcesoAMemoria
aniadirSiguienteProcesoACPU
asignarColorProceso
asignarColoresTabla
breakpoint
calcularPosProceso
calcularPosTodosProcesos
cargaDatos
colorearBarraTiempo
comprobacionDirectorio
comprobarDatosFichero
comprobarRango
comprobarRepite
comprobarSN
comprobarSiElProcesoEnCPUHaTerminado
convertirFicheroColorEnBlancoNegro
creacionDeDirectorios
datosAleatorios
datosFichero
datosManualProcesos
debug
deleteGeneratedFiles
dibujarEstadoCPU
dibujarMemoria
ejecucion
ejecutarUnCiloDeCPU
eliminarCola
eliminarPosProceso
eliminarProcesoDeMemoria
encontrarHuecoEnMemoria
escribeDatos
establecerPrioridad
```

```
finMain
fin_programa
forzarCierre
global
imprimirAviso
imprimirCabecera
imprimirErrorCritico
imprimirLCyan
imprimirTabla
imprimirTablaPredeterminada
inicializarArrays
main
medirTiempo
menuAlgoritmo
nularColumna
numAleatorio
ordenarProcesos
renombrarDatosEntrada
reubicarProcesos
salirPorErrorCritico
scanfNum
scanfNumMinMax
scanfSiNo
scanfString
selectorFichero
truncarBarraCPU
truncarMemoria
vaciarMemoria
```

DETAILS

Script Body

Has 7 line(s). Calls functions:

```
Script-Body
|-- comprobacionDirectorio
|   |-- imprimirLCyan
|   '-- salirPorErrorCritico
|       |-- forzarCierre
|       |   |-- deleteGeneratedFiles
|       |   |-- fin_programa
|       |   |   '-- imprimirLCyan
|       |   '-- imprimirErrorCritico
|       '-- imprimirErrorCritico
'-- global
    |-- creacionDeDirectorios
    |-- finMain
```

```

|-- abrirInforme
|-- `-- imprimirLCyan
|-- convertirFicheroColorEnBlancoNegro
|-- `-- renombrarDatosEntrada
|-- `-- scanfSiNo
|-- `-- imprimirAviso
|-- `-- imprimirLCyan
|-- main
|-- `-- cargaDatos
|-- `-- datosAleatorios
|-- `-- asignarColoresTabla
|-- `-- `-- asignarColorProceso
|-- `-- `-- imprimirErrorCritico
|-- `-- establecerPrioridad
|-- `-- numAleatorio
|-- `-- datosFichero
|-- `-- asignarColoresTabla
|-- `-- `-- asignarColorProceso
|-- `-- `-- imprimirErrorCritico
|-- `-- comprobarDatosFichero
|-- `-- `-- comprobarRango
|-- `-- `-- scanfNumMinMax
|-- `-- `-- imprimirAviso
|-- `-- `-- imprimirLCyan
|-- `-- `-- comprobarRepite
|-- `-- `-- scanfString
|-- `-- `-- imprimirAviso
|-- `-- `-- imprimirLCyan
|-- `-- `-- comprobarSN
|-- `-- `-- scanfSiNo
|-- `-- `-- imprimirAviso
|-- `-- `-- imprimirLCyan
|-- `-- establecerPrioridad
|-- `-- imprimirAviso
|-- `-- imprimirLCyan
|-- `-- scanfSiNo
|-- `-- `-- imprimirAviso
|-- `-- `-- imprimirLCyan
|-- `-- scanfString
|-- `-- `-- imprimirAviso
|-- `-- `-- imprimirLCyan
|-- `-- selectorFichero
|-- `-- `-- imprimirLCyan
|-- `-- `-- scanfNumMinMax
|-- `-- `-- imprimirAviso
|-- `-- `-- imprimirLCyan
|-- `-- datosManualProcesos
|-- `-- `-- asignarColorProceso
|-- `-- `-- imprimirErrorCritico
|-- `-- `-- imprimirTabla
|-- `-- `-- ordenarProcesos

```

```

|-- scanfNum
|   |-- imprimirAviso
|   |-- imprimirLCyan
|-- scanfNumMinMax
|   |-- imprimirAviso
|   |-- imprimirLCyan
|-- scanfSiNo
|   |-- imprimirAviso
|   |-- imprimirLCyan
|-- establecerPrioridad
|-- fin_programa
|   |-- imprimirLCyan
|-- menuAlgoritmo
|   |-- imprimirLCyan
|   |-- scanfNumMinMax
|   |-- imprimirAviso
|   |-- imprimirLCyan
|-- scanfNum
|   |-- imprimirAviso
|   |-- imprimirLCyan
|-- scanfNumMinMax
|   |-- imprimirAviso
|   |-- imprimirLCyan
|-- scanfSiNo
|   |-- imprimirAviso
|   |-- imprimirLCyan
|-- ejecucion
|   |-- anadirCola
|   |-- aniadirProcesoAMemoria
|   |   |-- ajustarMemoriaParaElProceso
|   |   |   |-- dibujarMemoria
|   |   |   |   |-- imprimirLCyan
|   |   |   |-- truncarMemoria
|   |   |-- encontrarHuecoEnMemoria
|   |   |-- imprimirAviso
|   |   |-- reubicarProcesos
|   |   |   |-- breakpoint
|   |   |-- vaciarMemoria
|   |-- breakpoint
|   |-- dibujarMemoria
|   |   |-- imprimirLCyan
|   |   |-- truncarMemoria
|   |-- eliminarCola
|   |-- imprimirErrorCritico
|   |-- salirPorErrorCritico
|   |   |-- forzarCierre
|   |   |   |-- deleteGeneratedFiles
|   |   |   |-- fin_programa
|   |   |   |-- imprimirLCyan
|   |   |-- imprimirErrorCritico
|   |-- imprimirErrorCritico

```

```

|-- aniadirSiguienteProcesoACPU
|   |-- breakpoint
|-- calcularPosTodosProcesos
|-- comprobarSiElProcesoEnCPUHaTerminado
|   |-- eliminarPosProceso
|   |-- eliminarProcesoDeMemoria
|-- dibujarEstadoCPU
|   |-- truncarBarraCPU
|   |-- colorearBarraTiempo
|-- dibujarMemoria
|   |-- imprimirLCyan
|   |-- truncarMemoria
|-- ejecutarUnCiloDeCPU
|-- imprimirLCyan
|-- imprimirTablaPredeterminada
|   |-- imprimirTabla
|-- vaciarMemoria
|-- escribeDatos
|   |-- ordenarProcesos
|-- imprimirLCyan
|-- imprimirTabla
|-- inicializarArrays
|-- nularColumna
|-- ordenarProcesos

```

DEV_ImprimirColores

```

# Nombre: DEV_ImprimirColores
# Date: 05/03/2020
# Descripción: Imprime por todos los procesos en el sistema: El color del string
y de fondo asignado

```

Has 13 line(s). Doesn't call other functions.

Uses feature(s): *read*

Not called by script or any function (may be e.g. `command_not_found_handle` or called indirectly in other way).

DEV_modificarMemoria

```

# Nombre: DEV_modificarMemoria
# Date: 27/01/2020
# Descripción: Función que permite manipular la memoria de forma manual. Diseñada
para testear el comportamiento de esta.

```

Has 44 line(s). Calls functions:

```
DEV_modificarMemoria
|-- aniadirProcesoAMemoria
|   |-- ajustarMemoriaParaElProceso
|   |   |-- dibujarMemoria
|   |   |   |-- imprimirLCyan
|   |   |   |-- truncarMemoria
|   |   |-- encontrarHuecoEnMemoria
|   |   |-- imprimirAviso
|   |   |-- reubicarProcesos
|   |       |-- breakpoint
|   |       |-- vaciarMemoria
|   |-- breakpoint
|   |-- dibujarMemoria
|   |   |-- imprimirLCyan
|   |   |-- truncarMemoria
|   |-- eliminarCola
|   |-- imprimirErrorCritico
|   |-- salirPorErrorCritico
|       |-- forzarCierre
|       |   |-- deleteGeneratedFiles
|       |   |-- fin_programa
|       |       |-- imprimirLCyan
|       |       |-- imprimirErrorCritico
|       |-- imprimirErrorCritico
|-- aniadirSiguienteProcesoACPU
|   |-- breakpoint
|-- eliminarProcesoDeMemoria
|-- imprimirAviso
|-- reubicarProcesos
|   |-- breakpoint
|   |-- vaciarMemoria
|-- scanfSiNo
|   |-- imprimirAviso
|   |-- imprimirLCyan
```

Uses feature(s): *read*

Not called by script or any function (may be e.g. `command_not_found_handle` or called indirectly in other way).

abrirInforme

```
# Nombre: abrirInforme
# Descripción: Función con distintas opciones para abrir el informe
# Date: 20/03/2020
```


Has 35 line(s). Calls functions:

```
abrirInforme  
'-- imprimirLCyan
```

Uses feature(s): *read*

Called by:

```
finMain
```

ajustarMemoriaParaElProceso

```
# Funcion que comprueba si un proces cabe en la memoria TOTAL, o es necesario  
reubicar.  
# Si es necesario, reubica la memoria.  
# Date: 22/02/2020  
# @param $1: Tamaño del proceso  
# @param $2/return: Variable en la que se almacenará el valor de salida de esta  
función  
# @return posición en la que empieza el hueco ó null si no hay suficiente hueco.  
# return por stdout, es necesario = la llamada de la función a una variable  
para "capturar" el return.
```

Has 13 line(s). Calls functions:

```
ajustarMemoriaParaElProceso  
|-- dibujarMemoria  
| |-- imprimirLCyan  
| '-- truncarMemoria  
|-- encontrarHuecoEnMemoria  
|-- imprimirAviso  
'-- reubicarProcesos  
    |-- breakpoint  
    '-- vaciarMemoria
```

Uses feature(s): *eval*

Called by:

```
aniadirProcesoAMemoria
```

anadirCola

```
# Nombre: anadirCola
# Descripcion: añade el indice de un proceso del array procesos al final de la
cola,
# cambia el estado del proceso a "cola"
# @param $1: indice del proceso a meter en cola
```

Has 4 line(s). Doesn't call other functions.

Called by:

ejecucion

aniadirProcesoAMemoria

```
# Si el proceso tiene un tamaño igual o menor a la memoria libre:
# -Se introduce el proceso en memoria
# -Se actualiza su estado a "STAT_MEMO"
# -Se actualiza la cantidad de memoria libre
# Date: 22/02/2020
# @param $1: indice del proceso a guardar en la particion
```

Has 33 line(s). Calls functions:

```

aniadirProcesoAMemoria
|-- ajustarMemoriaParaElProceso
|   |-- dibujarMemoria
|   |   |-- imprimirLCyan
|   |   |-- truncarMemoria
|   |-- encontrarHuecoEnMemoria
|   |-- imprimirAviso
|   |-- reubicarProcesos
|       |-- breakpoint
|       |-- vaciarMemoria
|-- breakpoint
|-- dibujarMemoria
|   |-- imprimirLCyan
|   |-- truncarMemoria
|-- eliminarCola
|-- imprimirErrorCritico
|-- salirPorErrorCritico
    |-- forzarCierre
    |   |-- deleteGeneratedFiles
    |   |-- fin_programa
    |   |-- imprimirLCyan
    |-- imprimirErrorCritico
    |-- imprimirErrorCritico

```

Called by:

```

DEV_modificarMemoria
ejecucion

```

aniadirSiguienteProcesoACPU

```

# Nombre: aniadirSiguienteProcesoACPU
# Date: 27/02/2020
# Descripción: De entre todos los procesos en memoria, añade el proces con la
prioridad más alta a CPU

```

Has 34 line(s). Calls functions:

```

aniadirSiguienteProcesoACPU
|-- breakpoint

```

Called by:

DEV_modificarMemoria
ejecucion

asignarColorProceso

```
# Nombre: asignarColorProceso.  
# Date: 05/03/2020  
# Descripción: Pasado el índice del proceso y un entero, se asignará a dicho  
proceso el color correspondiente al entero.  
# Ejemplo de uso: Al introducir un proceso manualmente, se le asigna el color al  
momento!  
# Nota: Los arrays de colores deben tener el mismo tamaño y los colores en la  
misma posición.  
# @Param $1: índice/puntero al proceso en la tabla procesos  
# @Param $2: entero cualquiera
```

Has 6 line(s). Calls functions:

```
asignarColorProceso  
\-- imprimirErrorCritico
```

Called by:

```
asignarColoresTabla  
datosManualProcesos
```

asignarColoresTabla

```
# Nombre: asignarColoresTabla  
# Date: 05/03/2020  
# Descripción: Rellena las columnas $P_COLOR y $P_COLORLETRA de la tabla procesos  
con colores  
# Ejemplo de uso: Al introducir los procesos por fichero/Random, podemos asignar  
a todos los procesos colores!
```

Has 5 line(s). Calls functions:

```
asignarColoresTabla  
\-- asignarColorProceso  
  \-- imprimirErrorCritico
```

Called by:

datosAleatorios
datosFichero

breakpoint

```
# Nombre: breakpoint
# Date: 21/02/2020
# Descripción: Permite realizar una parada del programa en cualquier punto del
código hasta que no se realizar una entrada por teclado.
# Uso: Activar o desactivar la variable global $BREAKPOINT_ENABLED para activar o
desactivar los breakpoints.
# Globales: BREAKPOINT_ENABLED
# @param $@: Imprime todos los stings pasados como argumento, por si se quieren
visualizar variables.
```

Has 14 line(s). Doesn't call other functions.

Uses feature(s): *read*

Called by:

aniadirProcesoAMemoria
aniadirSiguienteProcesoACPU
reubicarProcesos

calcularPosProceso

```
# Nombre: calcularPosProceso
# Descripción: Actualiza las variables $P_POSINI y $P_POSFIN del proceso indicado
# Date: 05/03/2020
# Ejemplo de USO: Si tuviese tiempo para hacer un programa eficiente, habría que
actualizar la tabla a paritr
#   de esta función cuando: Se añade un proceso en mem -> se reubica. Como no hay
tiempo y no se valora, se recalcula en cada bucle
#   la de todos los procesos en memoria.
# @Param $1: Índice del proceso a recalcular
#
#NOTA: Sin uso en todo el programa, no había presupuesto, posiblemente nunca se
haya probado.
```

Has 7 line(s). Doesn't call other functions.

Not called by script or any function (may be e.g. `command_not_found_handle` or called indirectly in other way).

calcularPosTodosProcesos

```
# Nombre: calcularPosTodosProcesos
# Descripción: Actualiza las variables $P_POSINI y $P_POSFIN de todos los
procesos que estén en memoria
# Date: 05/03/2020
```

Has 11 line(s). Doesn't call other functions.

Called by:

ejecucion

cargaDatos

```
# Nombre cargaDatos
# Descripcion: muestra las opciones de inicio del programa si los datos es por
fichero, teclado o automáticos
```

Has 38 line(s). Calls functions:

```
cargaDatos
|-- datosAleatorios
|   |-- asignarColoresTabla
|   |   '-- asignarColorProceso
|   |       '-- imprimirErrorCritico
|   |-- establecerPrioridad
|   '-- numAleatorio
|-- datosFichero
|   |-- asignarColoresTabla
|   |   '-- asignarColorProceso
|   |       '-- imprimirErrorCritico
|   |-- comprobarDatosFichero
|   |   |-- comprobarRango
|   |   |   '-- scanfNumMinMax
|   |   |       |-- imprimirAviso
|   |   |       '-- imprimirLCyan
|   |   |-- comprobarRepite
|   |   |   '-- scanfString
|   |   |       |-- imprimirAviso
|   |   |       '-- imprimirLCyan
|   |   '-- comprobarSN
|   |       '-- scanfSiNo
|   |           |-- imprimirAviso
|   |           '-- imprimirLCyan
```

```

|-- establecerPrioridad
|-- imprimirAviso
|-- imprimirLCyan
|-- scanfSiNo
|   |-- imprimirAviso
|   |-- imprimirLCyan
|-- scanfString
|   |-- imprimirAviso
|   |-- imprimirLCyan
|-- selectorFichero
|   |-- imprimirLCyan
|   |-- scanfNumMinMax
|       |-- imprimirAviso
|       |-- imprimirLCyan
|-- datosManualProcesos
|   |-- asignarColorProceso
|   |-- imprimirErrorCritico
|-- imprimirTabla
|-- ordenarProcesos
|-- scanfNum
|   |-- imprimirAviso
|   |-- imprimirLCyan
|-- scanfNumMinMax
|   |-- imprimirAviso
|   |-- imprimirLCyan
|-- scanfSiNo
|   |-- imprimirAviso
|   |-- imprimirLCyan
|-- establecerPrioridad
|-- fin_programa
|   |-- imprimirLCyan
|-- menuAlgoritmo
|   |-- imprimirLCyan
|   |-- scanfNumMinMax
|       |-- imprimirAviso
|       |-- imprimirLCyan
|-- scanfNum
|   |-- imprimirAviso
|   |-- imprimirLCyan
|-- scanfNumMinMax
|   |-- imprimirAviso
|   |-- imprimirLCyan
|-- scanfSiNo
|   |-- imprimirAviso
|   |-- imprimirLCyan

```

Called by:

datosFichero
main

colorearBarraTiempo

Has 13 line(s). Doesn't call other functions.

Called by:

truncarBarraCPU

comprobacionDirectorio

```
# Nombre: ComprobaciónDirectorio
# Date: 07/03/2020
# Descripción: Comprueba que el script está siendo ejecutándose desde ./algo.sh
# y no desde una dirección distinta (./test/priMayor/algo.sh),
# ya que crearía los archivos de salida en ./____/algo.sh y no en el mismo
directorio de ./algo.sh
```

Has 13 line(s). Calls functions:

```
comprobacionDirectorio
|-- imprimirLCyan
'-- salirPorErrorCritico
    |-- forzarCierre
    |   |-- deleteGeneratedFiles
    |   |-- fin_programa
    |   |-- '-- imprimirLCyan
    |   '-- imprimirErrorCritico
    '-- imprimirErrorCritico
```

Called by:

Script-Body

comprobarDatosFichero

```
# Nombre: comprobarDatosFichero
# Descripcion: comprueba si los datos del fichero son correctos. Si se repite el
nombre de los procesos, fuera de rango en tamaño de procesos etc.
```


Has 26 line(s). Calls functions:

```
comprobarDatosFichero
|-- comprobarRango
|   |-- scanfNumMinMax
|       |-- imprimirAviso
|       |-- imprimirLCyan
|-- comprobarRepite
|   |-- scanfString
|       |-- imprimirAviso
|       |-- imprimirLCyan
|-- comprobarSN
    |-- scanfSiNo
        |-- imprimirAviso
        |-- imprimirLCyan
```

Called by:

```
datosFichero
```

comprobarRango

```
# Nombre: comprobarRango
# Descripcion: comprueba número se encuentra dentro de un rango de dos números,
# si no se encuentra dentro llama a scanfNumMinMax para modificarlo
# @param $1: el texto que se imprime en caso de que el número no se encuentre
# dentro del rango o no sea un número
# @param $2: el número a comprobar si encuentra dentro del rango
# @param $3: variable a modificar si el numero no se encuentra dentro del rango
# @param $4: numero mínimo del rango incluido
# @param $5: numero máximo del rango incluido
```

Has 9 line(s). Calls functions:

```
comprobarRango
|-- scanfNumMinMax
    |-- imprimirAviso
    |-- imprimirLCyan
```

Called by:

```
comprobarDatosFichero
```

comprobarRepite

```
# Nombre: comprobarRepite
# Descripcion: comprueba el nombre del proceso se esta repitiendo
# @param $1: posicion del proceso del array procesos
```

Has 9 line(s). Calls functions:

```
comprobarRepite
`-- scanfString
    |-- imprimirAviso
    `-- imprimirLCyan
```

Called by:

```
comprobarDatosFichero
```

comprobarSN

```
# Nombre: comprobarSN
# Descripcion: comprueba si un número es un s o un n. Solo se usa en datos por
fichero
# @param $1: el texto a imprimir en pantalla en caso de valor no válido
# @param $2: valor de la variable a comprobar
# @param $3: variable a modificar en caso de valor no válido
```

Has 3 line(s). Calls functions:

```
comprobarSN
`-- scanfSiNo
    |-- imprimirAviso
    `-- imprimirLCyan
```

Called by:

```
comprobarDatosFichero
```

comprobarSiElProcesoEnCPUHaTerminado

```
# Nombre: comprobarSiElProcesoEnCPUHaTerminado
# Date: 29/02/2020
# Descripción: Si un proceso ha termiando en CPU -> Actualiza su línea en la
tabla y lo saca de CPU y memoria
```

Has 10 line(s). Calls functions:

```
comprobarSiElProcesoEnCPUHaTerminado
|-- eliminarPosProceso
`-- eliminarProcesoDeMemoria
```

Called by:

ejecucion

convertirFicheroColorEnBlancoNegro

```
# Nombre: convertirFicheroColorEnBlancoNegro
# Descripción: Lee un fichero formateado con ASCII Color Schemes y lo convierte
en Blanco y negro
#      Elimina los colores usados en este script y otros chars de escape para
poder ser visualizados en cualquier editor de texto
# Date: 29/02/2020
# Documentation: https://stackoverflow.com/questions/19296667/remove-ansi-color-codes-from-a-text-file-using-bash
# @Param $1: Direccion del fichero 1 a convertir
# @Param $2: Dirección del fichero 2 en el que se volcará el resultado
# @Param $3: String boolano ("true"), en el que se indica si se quiere borrar el
fichero original
```

Has 6 line(s). Doesn't call other functions.

Called by:

finMain

creacionDeDirectorios

```
# Nombre: creacionDeDirectorios()
# Descripción: Crea los directorios para los ficheros de entrada y los informes
si no existen
# Date: 21/03/2020
```

Has 7 line(s). Doesn't call other functions.

Called by:

global

datosAleatorios

```
# Nombre: datosAleatorios
# Descripción: Opción 3: Datos Aleatorios. Carga datos del número y tamaño de
partición,
# número de procesos, etc con números aleatorios llamando a la función
numAleatorio
```

Has 20 line(s). Calls functions:

```
datosAleatorios
|-- asignarColoresTabla
|   |-- asignarColorProceso
|       |-- imprimirErrorCritico
|-- establecerPrioridad
'-- numAleatorio
```

Called by:

cargaDatos

datosFichero

```
# Nombre: datosFichero
# Descripción: Opción 2: Por datos. Recoge todos los datos a través del fichero
```

Has 39 line(s). Calls functions:

```
datosFichero
|-- asignarColoresTabla
```

```

|-- asignarColorProceso
|   |-- imprimirErrorCritico
|-- cargaDatos
|   |-- datosAleatorios
|   |   |-- asignarColoresTabla
|   |   |   |-- asignarColorProceso
|   |   |   |-- imprimirErrorCritico
|   |   |-- establecerPrioridad
|   |   |-- numAleatorio
|   |-- datosManualProcesos
|   |   |-- asignarColorProceso
|   |   |   |-- imprimirErrorCritico
|   |   |-- imprimirTabla
|   |   |-- ordenarProcesos
|   |   |-- scanfNum
|   |   |   |-- imprimirAviso
|   |   |   |-- imprimirLCyan
|   |   |-- scanfNumMinMax
|   |   |   |-- imprimirAviso
|   |   |   |-- imprimirLCyan
|   |   |-- scanfSiNo
|   |   |   |-- imprimirAviso
|   |   |   |-- imprimirLCyan
|   |-- establecerPrioridad
|   |-- fin_programa
|   |   |-- imprimirLCyan
|   |-- menuAlgoritmo
|   |   |-- imprimirLCyan
|   |   |-- scanfNumMinMax
|   |   |   |-- imprimirAviso
|   |   |   |-- imprimirLCyan
|   |-- scanfNum
|   |   |-- imprimirAviso
|   |   |-- imprimirLCyan
|   |-- scanfNumMinMax
|   |   |-- imprimirAviso
|   |   |-- imprimirLCyan
|   |-- scanfSiNo
|   |   |-- imprimirAviso
|   |   |-- imprimirLCyan
|-- comprobarDatosFichero
|   |-- comprobarRango
|   |   |-- scanfNumMinMax
|   |   |   |-- imprimirAviso
|   |   |   |-- imprimirLCyan
|   |-- comprobarRepite
|   |   |-- scanfString
|   |   |   |-- imprimirAviso
|   |   |   |-- imprimirLCyan
|   |-- comprobarSN
|   |   |-- scanfSiNo

```

```

|           |-- imprimirAviso
|           '-- imprimirLCyan
|-- establecerPrioridad
|-- imprimirAviso
|-- imprimirLCyan
|-- scanfSiNo
|   |-- imprimirAviso
|   '-- imprimirLCyan
|-- scanfString
|   |-- imprimirAviso
|   '-- imprimirLCyan
'-- selectorFichero
    |-- imprimirLCyan
    '-- scanfNumMinMax
        |-- imprimirAviso
        '-- imprimirLCyan

```

Called by:

cargaDatos

datosManualProcesos

```

# Nombre: datosManualProcesos
# Descripcion: el usuario rellena los datos de cada proceso por teclado: nombre,
t.llegada, t.ejecución, prioridad y tamaño
# @param $1: índice del proceso dentro del array procesos

```

Has 31 line(s). Calls functions:

```

datosManualProcesos
|-- asignarColorProceso
|   '-- imprimirErrorCritico
|-- imprimirTabla
|-- ordenarProcesos
|-- scanfNum
|   |-- imprimirAviso
|   '-- imprimirLCyan
|-- scanfNumMinMax
|   |-- imprimirAviso
|   '-- imprimirLCyan
'-- scanfSiNo
    |-- imprimirAviso
    '-- imprimirLCyan

```

Called by:

cargaDatos

debug

```
# Nombre: debug (y sus muchas funciones)
# Date: 21/02/2020
# Descripción: Permite imprimir un string en un fichero a parte, y que este sea
visualizado desde otro terminal
# Uso: alternar los booleanos globales
# Globales:DEFAULT_DEBUG_OUTPUT_FILE_NAME, DEBUG_ENABLE, DEBUG_FIRST_EXECUTION,
DEBUG_PERSISTENT_FILE
# @param $1: String a imprimir
```

Has 27 line(s). Doesn't call other functions.

Not called by script or any function (may be e.g. command_not_found_handle or called indirectly in other way).

deleteGeneratedFiles

```
# Nombre: deleteGeneratedFiles
# Date: 21/02/2020
# Descripción: Borra aquellos archivos que han sido generados por el programa. Es
necesario indicar los archivos a borrar
# Uso: Añadir los archivos, pereferiblemente, comprobar antes de borrar si
existen
```

Has 7 line(s). Doesn't call other functions.

Called by:

forzarCierre

dibujarEstadoCPU

```
# Nombre: dibujarEstadoCPU
# Date: 09/03/2020
# Descripción: Imprime el estado de la CPU por pantalla
```

Has 23 line(s). Calls functions:

```
dibujarEstadoCPU
`-- truncarBarraCPU
   |-- colorearBarraTiempo
```

Called by:

```
ejecucion
```

dibujarMemoria

```
# Nombre: dibujarMemoria
# Descripción: Muestra por pantalla la memoria truncada
# @Param: $1 string de control: si el string es "mostrarStatsMemoria", se muestra
por pantalla el uso de memoria y el
```

Has 25 line(s). Calls functions:

```
dibujarMemoria
|-- imprimirLCyan
`-- truncarMemoria
```

Called by:

```
ajustarMemoriaParaElProceso
aniadirProcesoAMemoria
ejecucion
```

ejecucion

```
# Nombre: ejecucion
# Descripción: Loop central con la ejecución de los procesos
```

Has 94 line(s). Calls functions:


```

ejecucion
|-- anadirCola
|-- aniadirProcesoAMemoria
|   |-- ajustarMemoriaParaElProceso
|   |   |-- dibujarMemoria
|   |   |   |-- imprimirLCyan
|   |   |   |-- truncarMemoria
|   |   |-- encontrarHuecoEnMemoria
|   |   |-- imprimirAviso
|   |   |-- reubicarProcesos
|   |       |-- breakpoint
|   |       |-- vaciarMemoria
|   |-- breakpoint
|   |-- dibujarMemoria
|   |   |-- imprimirLCyan
|   |   |-- truncarMemoria
|   |-- eliminarCola
|   |-- imprimirErrorCritico
|   |-- salirPorErrorCritico
|       |-- forzarCierre
|       |   |-- deleteGeneratedFiles
|       |   |-- fin_programa
|       |       |-- imprimirLCyan
|       |       |-- imprimirErrorCritico
|       |-- imprimirErrorCritico
|-- aniadirSiguienteProcesoACPU
|   |-- breakpoint
|-- calcularPosTodosProcesos
|-- comprobarSiElProcesoEnCPUHaTerminado
|   |-- eliminarPosProceso
|   |-- eliminarProcesoDeMemoria
|-- dibujarEstadoCPU
|   |-- truncarBarraCPU
|   |-- colorearBarraTiempo
|-- dibujarMemoria
|   |-- imprimirLCyan
|   |-- truncarMemoria
|-- ejecutarUnCiloDeCPU
|-- imprimirLCyan
|-- imprimirTablaPredeterminada
|   |-- imprimirTabla
|-- vaciarMemoria

```

Uses feature(s): *read*

Called by:

```
main
```

ejecutarUnCicloDeCPU

```
# Nombre: ejecutarUnCicloDeCPU
# Date: 27/02/2020
# Descripción: Simula el comportamiento repetitivo de algunos momentos del
programa
#   -Calcula ciertos valores
#   -Aumenta el tiempo de ejecución
#   -Actualiza la línea de estado de CPU
```

Has 11 line(s). Doesn't call other functions.

Called by:

ejecucion

eliminarCola

```
# Nombre: eliminarCola
# Descripcion: elimina el primer elemento de la cola y mueve los demas elementos
a la izquierda
```

Has 6 line(s). Doesn't call other functions.

Called by:

aniadirProcesoAMemoria

eliminarPosProceso

```
# Nombre: eliminarPosProceso
# Date: 05/03/2020
# Descripción: Actualiza las variables $P_POSINI y $P_POSFIN del proceso
indicado a "-"
# @Param $1: índice del proces cuyas posiciones queremos eliminar.
```

Has 3 line(s). Doesn't call other functions.

Called by:

comprobarSiElProcesoEnCPUHaTerminado

eliminarProcesoDeMemoria

```
# Si el proceso se encuentra en memoria
# -Se elimina el proceso en memoria
# -Se actualiza su estado a "FIN"
# -Se actualiza la cantidad de memoria libre
# Date: 22/02/2020
# @param $1: indice del proceso a borrar de memoria
```

Has 14 line(s). Doesn't call other functions.

Called by:

```
DEV_modificarMemoria
comprobarSiElProcesoEnCPUHaTerminado
```

encontrarHuecoEnMemoria

```
# Nombre: encontrarHuecoEnMemoria
# Descripción: Función que calcula la posición / si es posible introducir $1
unidades de memoria continuas en memoria.
# Date 22/02/2020
# @param1 tamaño del hueco a encontrar / tamaño del proceso a emplazar
# @param $2/return: Variable en la que se almacenará el valor de salida de esta
función
# @return posición en la que empieza el hueco ó null si no hay suficiente hueco.
# return por stdout, es necesario = la llamada de la función a una variable
para "capturar" el return.
```

Has 35 line(s). Doesn't call other functions.

Uses feature(s): *eval*

Called by:

```
ajustarMemoriaParaElProceso
```

escribeDatos

```
# Nombre: escribeDatos
# Descripción: sobrescribe los datos en datos.txt, si no existe lo crea
```

Has 36 line(s). Calls functions:

```
escribeDatos  
'-- ordenarProcesos
```

Called by:

```
main
```

establecerPrioridad

```
# Nombre: establecerPrioridad  
# Descripción: establece el tipo de prioridad considerando priorMin y priorMax.  
El valor se usara para comparar en la ejecución  
# Globales: tipoPrioridad
```

Has 5 line(s). Doesn't call other functions.

Called by:

```
cargaDatos  
datosAleatorios  
datosFichero
```

finMain

```
# Nombre: finMain  
# Descripción: Es la función que se ejecuta tras el main (en global)  
# Todas las salidas de main se ven reflejadas en el archivo $INFORME_FILENAME,  
pero como no queremos todas, aislamos el final del programa  
# Nota: Si haces el | tee sobre algunas funciones de main, se rompe la ejecución,  
y si el finMain no se ejecuta desde global, se pierden los parámetros globales  
como colores o nombres de archivo
```

Has 3 line(s). Calls functions:

```
finMain
|-- abrirInforme
|  '-- imprimirLCyan
|-- convertirFicheroColorEnBlancoNegro
'-- renombrarDatosEntrada
    '-- scanfSiNo
        |-- imprimirAviso
        '-- imprimirLCyan
```

Called by:

global

fin_programa

```
# Nombre: fin_programa
# Descripcion: se termina la ejecución del script
```

Has 2 line(s). Calls functions:

```
fin_programa
'-- imprimirLCyan
```

Called by:

cargaDatos
forzarCierre

forzarCierre

```
# Nombre: forzarCierre
# Descripcion: Ejecuta una serie de comandos antes de salir, perfecta para borrar
archivo u otras cosas por si no es válida la ejecución!
# Date: 21/02/2020
# //@see deleteGeneratedFiles
```

Has 5 line(s). Calls functions:

```

forzarCierre
|-- deleteGeneratedFiles
|-- fin_programa
|   '-- imprimirLCyan
'-- imprimirErrorCritico

```

Uses feature(s): *read*

Called by:

```

salirPorErrorCritico

```

global

```

# Nombre: global
# Descripción: Es el bloque de código que alberga todas las variables globales
# ¿Por qué usar una función global?
#   No hay mucho beneficio más allá del estético, poder minimazar el bloque de
código de global (200 líneas aprox)
#   mejora muchísimo la navegabilidad del código.
#   Todas las funciones que quieran usar variables de global tienen que ser
llamdas desde: GLOBAL...
# Date: Pues no lo sé

```

Has 118 line(s). Calls functions:

```

global
|-- creacionDeDirectorios
|-- finMain
|   |-- abrirInforme
|   |   '-- imprimirLCyan
|   |-- convertirFicheroColorEnBlancoNegro
|   '-- renombrarDatosEntrada
|       '-- scanfSiNo
|           |-- imprimirAviso
|           '-- imprimirLCyan
'-- main
    |-- cargaDatos
    |   |-- datosAleatorios
    |   |   |-- asignarColoresTabla
    |   |   |   '-- asignarColorProceso
    |   |   |       '-- imprimirErrorCritico
    |   |   |-- establecerPrioridad
    |   |   '-- numAleatorio
    |   |-- datosFichero
    |   |-- asignarColoresTabla

```

```

|-- asignarColorProceso
|-- imprimirErrorCritico
|-- comprobarDatosFichero
|-- comprobarRango
|-- scanfNumMinMax
|-- imprimirAviso
|-- imprimirLCyan
|-- comprobarRepite
|-- scanfString
|-- imprimirAviso
|-- imprimirLCyan
|-- comprobarSN
|-- scanfSiNo
|-- imprimirAviso
|-- imprimirLCyan
|-- establecerPrioridad
|-- imprimirAviso
|-- imprimirLCyan
|-- scanfSiNo
|-- imprimirAviso
|-- imprimirLCyan
|-- scanfString
|-- imprimirAviso
|-- imprimirLCyan
|-- selectorFichero
|-- imprimirLCyan
|-- scanfNumMinMax
|-- imprimirAviso
|-- imprimirLCyan
|-- datosManualProcesos
|-- asignarColorProceso
|-- imprimirErrorCritico
|-- imprimirTabla
|-- ordenarProcesos
|-- scanfNum
|-- imprimirAviso
|-- imprimirLCyan
|-- scanfNumMinMax
|-- imprimirAviso
|-- imprimirLCyan
|-- scanfSiNo
|-- imprimirAviso
|-- imprimirLCyan
|-- establecerPrioridad
|-- fin_programa
|-- imprimirLCyan
|-- menuAlgoritmo
|-- imprimirLCyan
|-- scanfNumMinMax
|-- imprimirAviso
|-- imprimirLCyan

```

```

|-- scanfNum
|   |-- imprimirAviso
|   '-- imprimirLCyan
|-- scanfNumMinMax
|   |-- imprimirAviso
|   '-- imprimirLCyan
|-- scanfSiNo
|   |-- imprimirAviso
|   '-- imprimirLCyan
|-- ejecucion
|   |-- anadirCola
|   |-- aniadirProcesoAMemoria
|   |   |-- ajustarMemoriaParaElProceso
|   |   |   |-- dibujarMemoria
|   |   |   |   |-- imprimirLCyan
|   |   |   '-- truncarMemoria
|   |   |-- encontrarHuecoEnMemoria
|   |   |-- imprimirAviso
|   |   '-- reubicarProcesos
|   |       |-- breakpoint
|   |       '-- vaciarMemoria
|   |-- breakpoint
|   |-- dibujarMemoria
|   |   |-- imprimirLCyan
|   |   '-- truncarMemoria
|   |-- eliminarCola
|   |-- imprimirErrorCritico
|   '-- salirPorErrorCritico
|       |-- forzarCierre
|       |   |-- deleteGeneratedFiles
|       |   |-- fin_programa
|       |   |   '-- imprimirLCyan
|       |   '-- imprimirErrorCritico
|       '-- imprimirErrorCritico
|-- aniadirSiguienteProcesoACPU
|   '-- breakpoint
|-- calcularPosTodosProcesos
|-- comprobarSiElProcesoEnCPUHaTerminado
|   |-- eliminarPosProceso
|   '-- eliminarProcesoDeMemoria
|-- dibujarEstadoCPU
|   '-- truncarBarraCPU
|       '-- colorearBarraTiempo
|-- dibujarMemoria
|   |-- imprimirLCyan
|   '-- truncarMemoria
|-- ejecutarUnCiloDeCPU
|-- imprimirLCyan
|-- imprimirTablaPredeterminada
|   '-- imprimirTabla
|-- vaciarMemoria

```



```
|-- escribeDatos  
|  '-- ordenarProcesos  
|-- imprimirLCyan  
|-- imprimirTabla  
|-- inicializarArrays  
|-- nularColumna  
|-- ordenarProcesos
```

Called by:

Script-Body

imprimirAviso

```
# Nombre: imprimirAviso  
# Descripcion: imprime en pantalla un aviso de error al introducir un dato con  
letras.  
# @param $1: texto de aviso  
# Cambios 2020: Como hemos añadido la funcion imprimirErrorCritico, he cambiado  
el color de fondo de ROJO a AMARILLO/NARANJA (Depende del terminal)
```

Has 1 line(s). Doesn't call other functions.

Called by:

```
DEV_modificarMemoria  
ajustarMemoriaParaElProceso  
datosFichero  
scanfNum  
scanfNumMinMax  
scanfSiNo  
scanfString
```

imprimirCabecera

```
# Nombre: imprimirCabecera  
# Descripción: Imprime la cabecera del programa  
# Date: 7/03/2020
```

Has 26 line(s). Doesn't call other functions.

Not called by script or any function (may be e.g. `command_not_found_handle` or called indirectly in other way).

imprimirErrorCritico

```
# Nombre: imprimirErrorCritico
# Descripcion: imprime en pantalla un aviso de error al introducir un dato con
letras.
# @param $1: texto de aviso
```

Has 1 line(s). Doesn't call other functions.

Called by:

```
aniadirProcesoAMemoria
asignarColorProceso
forzarCierre
salirPorErrorCritico
```

imprimirLCyan

```
# Nombre: imprimirLCyan
# Descripcion: imprime en pantalla el text de color cyan claro
# @param $1: texto a imprimir en cyan claro
# @param $2 (opcional): argumento adicional para el echo como -n si no se quiere
introducir un salto de linea
```

Has 1 line(s). Doesn't call other functions.

Called by:

```
abrirInforme
comprobacionDirectorio
datosFichero
dibujarMemoria
ejecucion
fin_programa
main
menuAlgoritmo
scanfNum
scanfNumMinMax
scanfSiNo
scanfString
selectorFichero
```

imprimirTabla

```
# Nombre: imprimirTabla
# Descripcion: imprime las columnas del array procesos pasado como parámetro
# Versión 2.0 (Original imprimirTablaOld())
# Date: 6/03/2020
# Nota 2020: Adaptada a los requisitos exigidos en el curso 18-19 y 19-20 (Tabla compacta roñosa)
# @param $@ (todos): índice de las columnas que se quiere imprimir en pantalla
```

Has 23 line(s). Doesn't call other functions.

Called by:

```
datosManualProcesos
imprimirTablaPredeterminada
main
```

imprimirTablaPredeterminada

```
# Nombre: imprimirTablaPredeterminada
# Date: 06/03/2020
# Descripción: Función que llama a la función "imprimirTabla()" con unos parámetros específicos
#     Además, la función imprime una cabecera inicial.
# Nota: La tabla original era muy grande, y permitía imprimir las columnas seleccionadas.
#     En la versión de 2020 se exige que se impriman casi todas las columnas del struct, y tener que hacer una llamada
#     con todos los parámetros, puede ser muy pesado.
#     He decidido hacer esto por dos razones: 1º: Puedo reutilizar la tabla en la entrada de datos con 4 columnas 2º: Puedo poner la cabecera fija cómodamente.
```

Has 5 line(s). Calls functions:

```
imprimirTablaPredeterminada
`-- imprimirTabla
```

Called by:

```
ejecucion
```

inicializarArrays

Nombre: inicializarArrays
Descripción: inicializa arrays necesario antes de la ejecución

Has 10 line(s). Doesn't call other functions.

Called by:

main

main

#main

Has 27 line(s). Calls functions:

```
main
|-- cargaDatos
|   |-- datosAleatorios
|   |   |-- asignarColoresTabla
|   |   |   '-- asignarColorProceso
|   |   |       '-- imprimirErrorCritico
|   |   |-- establecerPrioridad
|   |   '-- numAleatorio
|   |-- datosFichero
|   |   |-- asignarColoresTabla
|   |   |   '-- asignarColorProceso
|   |   |       '-- imprimirErrorCritico
|   |   |-- comprobarDatosFichero
|   |   |   |-- comprobarRango
|   |   |   |   '-- scanfNumMinMax
|   |   |   |       |-- imprimirAviso
|   |   |   |       '-- imprimirLCyan
|   |   |   |-- comprobarRepite
|   |   |   |   '-- scanfString
|   |   |   |       |-- imprimirAviso
|   |   |   |       '-- imprimirLCyan
|   |   |   '-- comprobarSN
|   |   |       '-- scanfSiNo
|   |   |           |-- imprimirAviso
|   |   |           '-- imprimirLCyan
|   |   |-- establecerPrioridad
|   |   |-- imprimirAviso
|   |   '-- imprimirLCyan
```

```

|-- scanfSiNo
|   |-- imprimirAviso
|   |-- imprimirLCyan
|   |-- scanfString
|   |-- imprimirAviso
|   |-- imprimirLCyan
|   |-- selectorFichero
|   |-- imprimirLCyan
|   |-- scanfNumMinMax
|   |-- imprimirAviso
|   |-- imprimirLCyan
|-- datosManualProcesos
|   |-- asignarColorProceso
|   |-- imprimirErrorCritico
|   |-- imprimirTabla
|   |-- ordenarProcesos
|   |-- scanfNum
|   |-- imprimirAviso
|   |-- imprimirLCyan
|   |-- scanfNumMinMax
|   |-- imprimirAviso
|   |-- imprimirLCyan
|   |-- scanfSiNo
|   |-- imprimirAviso
|   |-- imprimirLCyan
|-- establecerPrioridad
|-- fin_programa
|   |-- imprimirLCyan
|-- menuAlgoritmo
|   |-- imprimirLCyan
|   |-- scanfNumMinMax
|   |-- imprimirAviso
|   |-- imprimirLCyan
|-- scanfNum
|   |-- imprimirAviso
|   |-- imprimirLCyan
|-- scanfNumMinMax
|   |-- imprimirAviso
|   |-- imprimirLCyan
|-- scanfSiNo
|   |-- imprimirAviso
|   |-- imprimirLCyan
|-- ejecucion
|   |-- anadirCola
|   |-- aniadirProcesoAMemoria
|   |-- ajustarMemoriaParaElProceso
|   |-- dibujarMemoria
|   |-- imprimirLCyan
|   |-- truncarMemoria
|   |-- encontrarHuecoEnMemoria
|   |-- imprimirAviso

```

```

|-- reubicarProcesos
|   |-- breakpoint
|   |-- vaciarMemoria
|-- breakpoint
|-- dibujarMemoria
|   |-- imprimirLCyan
|   |-- truncarMemoria
|-- eliminarCola
|-- imprimirErrorCritico
|-- salirPorErrorCritico
|   |-- forzarCierre
|   |-- deleteGeneratedFiles
|   |-- fin_programa
|   |-- `-- imprimirLCyan
|   |-- `-- imprimirErrorCritico
|   |-- `-- imprimirErrorCritico
|-- aniadirSiguienteProcesoACPU
|   |-- breakpoint
|-- calcularPosTodosProcesos
|-- comprobarSiElProcesoEnCPUHaTerminado
|   |-- eliminarPosProceso
|   |-- `-- eliminarProcesoDeMemoria
|-- dibujarEstadoCPU
|   |-- truncarBarraCPU
|   |-- `-- colorearBarraTiempo
|-- dibujarMemoria
|   |-- imprimirLCyan
|   |-- `-- truncarMemoria
|-- ejecutarUnCiloDeCPU
|-- imprimirLCyan
|-- imprimirTablaPredeterminada
|   |-- `-- imprimirTabla
|-- `-- vaciarMemoria
|-- escribeDatos
|-- `-- ordenarProcesos
|-- imprimirLCyan
|-- imprimirTabla
|-- inicializarArrays
|-- nularColumna
|-- `-- ordenarProcesos

```

Uses feature(s): *read*

Called by:

global

medirTiempo

Has 4 line(s). Doesn't call other functions.

Not called by script or any function (may be e.g. `command_not_found_handle` or called indirectly in other way).

menuAlgoritmo

```
# Nombre: menuAlgoritmo
# Descripcion: menu para elegir las opciones del algoritmo, más info en la opcion
5 Ayuda (glosario)
```

Has 44 line(s). Calls functions:

```
menuAlgoritmo
|-- imprimirLCyan
`-- scanfNumMinMax
    |-- imprimirAviso
    `-- imprimirLCyan
```

Uses feature(s): *read*

Called by:

```
cargaDatos
```

nularColumna

```
# Nombre: nularColumna
# Date: 05/03/2020
# Descripción: Actualiza toda una columna del STRUCT/${procesos[]} con el valor
"_"
# Ejemplo de uso: Nulado inicial de las columnas que no han sido introducidas
mediante la entrada.
# @Param $@: Todos los enteros/valores del struct que queremos vaciar
```

Has 14 line(s). Doesn't call other functions.

Called by:

```
main
```

numAleatorio

```
# Nombre: numAleatorio
# Descripcion: genera un numero aleatorio de un rango de dos numeros pasado como parametro
# @param $1: variable a asignar el número aleatorio
# @param $2: número mmáximo del rango incluido
# @param $3: número máximo del rango incluido
```

Has 9 line(s). Doesn't call other functions.

Uses feature(s): *eval*

Called by:

```
datosAleatorios
```

ordenarProcesos

```
# Nombre: ordenarProcesos
# Descripcion: ordena el array procesos en función del tiempo de llegada.
# Se usa selection sort como algortimo de ordenamiento
# Version 2.0
# Date 17/03/2020
# Detalles de modificación: Si dos prioridades son iguales, el que haya sido
# introducido antes tiene prioridad
#      EJ: P01 entra antes que P03, aunque tengan el mismo t.llegada.
```

Has 22 line(s). Doesn't call other functions.

Called by:

```
datosManualProcesos
escribeDatos
main
```

renombrarDatosEntrada

```
# Nombre: renombrarDatosEntrada
# Descripción: Renombra el archivo datos.txt para que los datos no sean borrados
# en la proxima ejecución
# Date: 19/03/2020
```


Has 17 line(s). Calls functions:

```
renombrarDatosEntrada
`-- scanfSiNo
   |-- imprimirAviso
   `-- imprimirLCyan
```

Uses feature(s): *read*

Called by:

```
finMain
```

reubicarProcesos

```
# Nombre: reubicarProcesos
# date: 22/03/2020
# Descripción: Reubica la memoria
# Nota del autor: Si no hubiese hecho el array bidimensional, esto podría haber
sído un simple oneLiner que ordenase el array de menor a mayor
# (No he tenido en cuenta donde irían los nulls, pero podrían haber sido
sustiutidos por 0 )
# indepentientemente, bash es una chusta, e igual no va tan bien como debería.
```

Has 25 line(s). Calls functions:

```
reubicarProcesos
|-- breakpoint
`-- vaciarMemoria
```

Called by:

```
DEV_modificarMemoria
ajustarMemoriaParaElProceso
```

salirPorErrorCritico

```
# Nombre: salirPorErroCritico
# Descripcion: imprime en pantalla un aviso de error al introducir un dato con
letras y para la ejecucion.
# @param $1: texto de aviso
# Date 21/02/2020
#//@see ErrorCritio
#//@see forzarCierre
```

Has 2 line(s). Calls functions:

```
salirPorErrorCritico
|-- forzarCierre
|   |-- deleteGeneratedFiles
|   |-- fin_programa
|   |   '-- imprimirLCyan
|   '-- imprimirErrorCritico
'-- imprimirErrorCritico
```

Called by:

```
aniadirProcesoAMemoria
comprobacionDirectorio
```

scanfNum

```
# Nombre: scanfNum
# Descripcion: asigna un valor mayor que un numero pasado como parametro a una
variable desde el teclado
# @param $1: texto a imprimir en pantalla para pedir
# @param $2: variable al que se asigna el numero valido
# @param $3: numero minimo valido
```

Has 9 line(s). Calls functions:

```
scanfNum
|-- imprimirAviso
'-- imprimirLCyan
```

Uses feature(s): *eval, read*

Called by:

```
cargaDatos
datosManualProcesos
```

scanfNumMinMax

```
# Nombre: scanfNumMinMax
# Descripcion: asigna un valor entre un rango de numeros desde el teclado
# @param $1: texto a imprimir para pedir el numero
# @param $2: variable al que se asigna el numero valido
# @param $3: numero mínimo del rango (incluido)
# @param $4: numero máximo del rango (incluido)
```

Has 16 line(s). Calls functions:

```
scanfNumMinMax
|-- imprimirAviso
'-- imprimirLCyan
```

Uses feature(s): *eval, read*

Called by:

```
cargaDatos
comprobarRango
datosManualProcesos
menuAlgoritmo
selectorFichero
```

scanfSiNo

```
# Nombre: scanfSiNo
# Descripcion: asigna un valor valido a una variable desde el teclado
# @param $1: texto a imprimir en pantalla para pedir si o no
# @param $2: variable al que se asigna si o no
```

Has 9 line(s). Calls functions:

```
scanfSiNo
|-- imprimirAviso
'-- imprimirLCyan
```

Uses feature(s): *eval, read*

Called by:

```
DEV_modificarMemoria  
cargaDatos  
comprobarSN  
datosFichero  
datosManualProcesos  
renombrarDatosEntrada
```

scanfString

```
# Nombre: scanfString  
# Descripcion: asigna un valor cadena de caracteres a una variable pasado como  
parametro  
# @param $1: texto a imprimir para pedir la cadena de caracteres  
# @param $2: variable al que se asigna la cadena valida
```

Has 12 line(s). Calls functions:

```
scanfString  
|-- imprimirAviso  
'-- imprimirLCyan
```

Uses feature(s): *eval, read*

Called by:

```
comprobarRepite  
datosFichero
```

selectorFichero

```
# Nombre: selectorFichero
# Descripción: muestra por pantalla los ficheros correspondientes a la secuencia
de escape $1
# y devuelve el string del fichero obtenido
# Nota: Esta función utiliza un fichero temporal que se almacena en /tmp
# Se requiere de la función mktemp que no es POSIX. Está instalada en muchos
sistemas, pero ojo cuidado!
#
# --NO-----@Param $1: secuencia de escape: por ejemplo $1='*.txt' muestra sólo
los ficheros que terminen con 'txt', es el resultado de un ls-----
# @Param $2: variable en la que vamos a almacenar el string de resultado
# GLOBAL: los valores se asignan también a $nomFile, ya que eval no permite hacer
return de un string con espacios
# Paso de comerme la cabeza, es una chapuza pero no hay otra, bash tiene
muchas limitaciones en este aspecto.
# Date: 19/03/2020
```

Has 19 line(s). Calls functions:

```
selectorFichero
|-- imprimirLCyan
'-- scanfNumMinMax
    |-- imprimirAviso
    '-- imprimirLCyan
```

Uses feature(s): *eval*

Called by:

```
datosFichero
```

truncarBarraCPU

```
# Nombre: truncarBarraCPU
# Date: 09/03/2020
# Descripción: Trunca la barra de CPU, y añade el tiempo de inicio y fin de cada
proceso, así como el nombre correspondiente.
# Nota: Tiene un uso similar al truncado de memoria. Los cálculos de las
distintas variables aquí referenciadas han sido realizados en dibujarEstadoCPU()
# ya que es necesario saber de antemano el ancho y alto del array a
imprimir y a generar, y puede variar si se calcula 2 veces, una en cada función.
# Además, aquí no tenemos un vector.length para saber cuanto mide un
array, bash-ura de lenguaje.
# Podría hacerse en una misma función, pero los parámetros en Bash son un
dolor, y paso de perder el tiempo haciendo un código de muy alta calidad.
```

Has 40 line(s). Calls functions:

```
truncarBarraCPU  
'-- colorearBarraTiempo
```

Called by:

```
dibujarEstadoCPU
```

truncarMemoria

```
# Nombre: truncarMemoria  
# Date: 08/03/2020  
# Parámetros: Utiliza la variable memoriaTruncada declarada en dibujarMemoria  
# Descripción: rellena un array bidimensional con el string de memoria a  
imprimir.  
#           este nuevo string permite representar la memoria de forma dinámica,  
dependiendo del ancho del terminal.
```

Has 37 line(s). Doesn't call other functions.

Called by:

```
dibujarMemoria
```

vaciarMemoria

```
# Nombre: vaciarMemoria  
# Date: 22/02/2020  
# Funcionamiento: Vacía la memoria segúnNecesidades, o la pone en su estado por  
defecto.
```

Has 4 line(s). Doesn't call other functions.

Called by:

```
ejecucion  
reubicarProcesos
```