## Proceso

prioridad : Entero

tiempoEjecucion : Entero

tiempoEjecutado : Entero

periodoInterrupcionES : Entero

largoInterrupcionES : Entero

id : String

estados : static enum

estadoActual : estados

getEstadoActual:

Cuando a un proceso se le solicita su estado actual, este verifica sus atributos para deducir en que estado debe estar, si debía cambiar lo hace y devuelve el estado en el que queda. El método funciona en simultaneo como un “updateState”, incluso si no se persiste el estado.

PreCondiciones:

Al proceso se le asignaron sus atributos de manera correcta y no nula

El proceso cuenta con un enumerador estático que representa los diferentes estados posibles

PostCondiciones:

El proceso actualiza su estado, de ser necesario

El proceso devuelve su estado actual

Pseudo:

Proceso.getEstadoActual() : devuelve Proceso.Estado

COM

SI tiempoEjecucion = tiempoEjecutado

This.estado <- Finalizado

Devuelve finalizado

SINO SI tiempoEjecutado mod periodoInterrupcionES = 0

This.estado <- BloqueadoES

Devuelve BloqueadoES

SINO

Devuelve this.Estado

FINSI

FIN

## Planificador

listaListos : LinkedList<Proceso>[99]

listaBloqueados : LinkedList<Proceso>

quantum : Entero

despachar(Proceso proc) : void

revisarBloqueados() : void

## CPU

procesoCargado : Proceso

ejecutarProceso(Proceso proc) : proc.EstadoActual

### ejecutarProceso:

Cuando el planificador le despacha al CPU un proceso, este simula ejecutarlo sumando al atributo de tiempoEjecutado del mismo, de manera secuencial y verificando si en algún momemento se bloquea el mismo por si solo o si finaliza. Una vez que ejecuto todos los ciclos que le corresponden, devuelve el estado final en el que quedo el proceso.

Precondiciones:

Hay un quantum definido

EL CPU tiene un Proceso Cargado

El proceso esta listo para ejecución

Postcondiciones:

El proceso tiene “quantum” mas en su tiempo ejecutado

A lo largo de la ejecución, el proceso puede bloquearse o finalizar, por lo que su ejecución termina

Devuelve el estado resultante del proceso

Pseudo:

CPU.ejecutarProceso() : devuelve Proceso.Estado

COM

Entero quantum

Para cada i desde 0 hasta quantum

Proceso.tiempoCargado++

Estado <- Proceso.getEstadoActual()

Si Estado = Finalizado o Estado = BloqueadoES

Devuelve estado

FINSI

FIN PARACADA

Devuelve Proceso.getEstadoActual()

FIN