

Tarea 5

Jorge Chavarría Herrera B82073

Sergio Martinez Calvo B84621

Rodrigo Vilchez Ulloa B78292

- 1) Un hilo para el procesador , un hilo para el buffer víctima.
- 2)

Quantum := variable

Instrucción := 0

Inmediato := 0

RegistroBase := 0

RegistroFuente := 0

mutex bloquesBuffer[8]

bloqueAEditar=0

contadorBuffer=0

relojNucleo=0

relojBuffer=0

mutex cicloReloj

mutex clcloBuffer

Hililo toma el nucleo

Repetir mientras que Quantum > 0:

Instrucción = r0

Fetch(Instrucción)

Fetch(Instrucción):

Si instrucción == 37:

Inmediato := r1

RegistroBase := r2

RegistroFuente := r3

direccion = (RegistroFuente + Inmediato)

NumeroBloque = direccion // 8

Si direccion % 8 == 0:

NumeroPalabra = 0

Sí no:

NumeroPalabra = 1

Si EscribirCache (dirección) == falso:

RevisarBuffer(getBloqueCaché(Direccion))

relojNucleo+=4

Quantum--

relojNucleo++

cicloReloj.wait();

Si relojNucleo>=relojBuffer :

relojBufer=relojNucleo

```
cicloBuffer.signal()
cicloReloj.signal();
```

EscribirCache (Dirección):

```
    contador := 0
    Repetir hasta que contador == 8:
        Si estado del bloque == "C":
            Por cada palabra del bloque:
                Si direcciónDePalabraEnBloqueCache(palabra) == Dirección:
                    palabra = RegistroBase
                    estado del bloque := "M"
                    return verdadero
    return falso
```

getBloqueCaché(Dirección):

```
    contador := 0
    Repetir hasta que contador == 8:
        Si estado del bloque == "C":
            Por cada palabra del bloque:
                Si direcciónDePalabraEnBloqueCache(palabra) == Dirección:
                    return cache[contador].bloque
    return -1
```

RevisarBuffer(Bloque):

```
    mismoBloque=falso

    Repetir hasta que contador == 8:
        bloquesBuffer[contador].wait()
        si bloque[2] == getNúmeroBloqueBuffer(contador):
            contadorBuffer=contador
            mismoBloque=verdadero
        bloquesBuffer[contador].signal()

    Si mismoBloque==falso:
        contadorBuffer := bloqueAEditar
        bloqueAEditar=(bloqueAEditar+1)%8

    Repetir hasta que listo == verdadero:
        bloquesBuffer[contadorBuffer].wait()
        Si getEstadoBloqueBuffer(contadorBuffer)=="Vacio":
            setDatosBuffer(contadorBuffer, Bloque)
            setEstadoBloqueBuffer("Cola")
            listo:=verdadero
    Si no:
```

Si mismoBloque==falso::

Si getEstadoBloqueBuffer(contadorBuffer)=="Cola":
setDatosBuffer(contadorBuffer, Bloque)

Si no:

siguiente=bloqueAEditar
bloqueAEditar=(bloqueAEditar+1)%8

bloquesBuffer[contadorBuffer].signal
contadorBuffer = siguiente

Clase Buffer:

getEstadoBloqueBuffer(contadorBuffer)
setDatosBuffer(contadorBuffer, Bloque)
setDatosBuffer(contadorBuffer, Bloque)

Casos Buffer

1. Caso buffer tiene campo: el núcleo le copia al buffer y luego el buffer pasa a memoria.
2. Caso buffer no tiene espacio: Bloques no se han pasado a memoria (todos en proceso de pasarse a memoria). Núcleo se espera.
3. Caso núcleo necesita escribir sobre un bloque que está en el buffer y no se está copiando: merging buffer.
4. Caso necesito escribir sobre un bloque que está en el buffer y se está copiando: lo escribo en otro bloque (casos 1 o 2).
5. Necesito leer de memoria y el buffer está pasando: copio del buffer a mi cache.