Project 2: Second Deliverable

Carolina Rojano, David Meléndez, Juan Luis Ávila

Escuela de Ingeniería, Ciencia y Tecnología, Universidad del Rosario

May 02 of 2024

For Pseudo-code, code, and class diagram refer to the following link: https://github.com/C9rol1ne/OperatingSystems/tree/master

Paging

Segmentation

For the Segmentation implementation 3 new methods were implemented, as helper methods in the computation of the physical address. Those new methods were:

- getCumulativeLogicalAddressSpace
- getSegmentNumber
- getOffset

The getCumulativeLogicalAddressSpace calculates the cumulative logical address space for a segment table by returning it as an ArrayList object.

The getSegmentNumber method receives a logical address and calculates its corresponding segment number.

The getOffset method takes a segment number and a logical address and calculates the corresponding offset.

Finally, the getPhysicalAddress method utilizes the previously defined helper methods to determine the segment number and offset of the logical address. It then retrieves the base address and limit of the corresponding segment from the segment table and checks if the offset is within the segment's limit. If so, it computes the physical address by adding the offset to the base address.

```
1: function getSegmentNumber(logAdd)
2:     for i ← 0 to st.getSize() − 1 do
3:         if logAdd < cumulativeLogAddSpace[i] then
4:             return i
5:         end if
6:     end for
7:     return −1                                   ▷ handle exception
8: end function
```

```
1: function getOffset(segmentNumber, logAdd)
2:     logicalBase ← 0
3:     offset ← 0
4:     if segmentNumber ≠ 0 then
5:         logicalBase ← cumulativeLogAddSpace[segmentNumber − 1]
6:     end if
7:     offset ← logAdd − logicalBase
8:     return offset
9: end function
```

```
1: function getCumulativeLogicalAddressSpace(S)
2:     CLAS ← new ArrayList
3:     limit ← st.getSegment(0).getLimit()
4:     CLAS.add(limit)                                    ▷ initialize CLAS
5:     for i ← 1 to st.getSize() − 1 do
6:         segment ← st.getSegment(i)
7:         length ← segment.getLimit()
8:         clas ← CLAS.get(i − 1)
9:         CLAS.add(length + clas)
10:    end for
11:    return CLAS
12: end function
```

```
1: function getPhysicalAddress(logicalAddress)
2:     segmentNumber ← getSegmentNumber(logicalAddress)
3:     offset ← getOffset(segmentNumber, logicalAddress)
4:     physicalAdd ← 0
5:     segment ← st.getSegment(segmentNumber)
6:     if offset < segment.getLimit() then
7:         physicalAdd ← offset + segment.getBase()
8:     else
9:         throw  IllegalArgumentException("offset is greater than limit")
10:    end if
11:    return physicalAdd
12: end function
```