TASK 4: ALBAN BERTHELOT

LOCATION OF THE PAGE REPLACEMENT ALGORITHM

This algorithm is located in the repository “work/” more precisely in the file named “pageFault.java”.

DESCRIPTION OF THE PAGE REPLACEMENT ALGORITHM

“This method gets called whenever a page needs to be replaced.” (cf. pageFault.java)
The goal of this algorithm is to link a physical page to an unlinked virtual page.

The first loop of the program is a “while” loop. It is used to search the oldest physical page linked. There are two solutions to exit this loop:

- Find the oldest page (variable “mapped == true”)
- Test all pages

In this loop, we can find several “if” loops.

1. We check if the current page is linked (i.e. “page.physical != -1”). If the condition is checked, we have to possible cases (so two “if” loop).
    a. If it is the first loop/test (i.e. “firstPage == -1”) we set the variable “firstPage = count”. That mean we set the first page to test to the first position of the vector “mem”.
    b. After this test, we check if the memory time corresponding to the actual page (variable “inMemTime”) is bigger than the oldest time (variable “oldestTime”) registered. If yes, the oldest time is replaced by the time of the current page (i.e. “oldestTime = page.inMemTime) and the oldest page is set to the actual page (i.e. “oldestPage = count”). To finish, the variable “mapped” switch to “true” (to exit the “while” loop).
2. After that we increase “count” by one to access the next page.
3. If “count” is equal to the number of virtual page (i.e. “count == virtPageNum”), it means that we have test all page and we have not found the oldest page, we must exit the “while” loop so “mapped” is set to “true”.
4. After the “while” if we are in the case number 3 (“oldestPage == -1”), an “if” loop is used to map the oldest page with the first page linked (“oldestPage = firstPage”).

For example, with 3 pages:

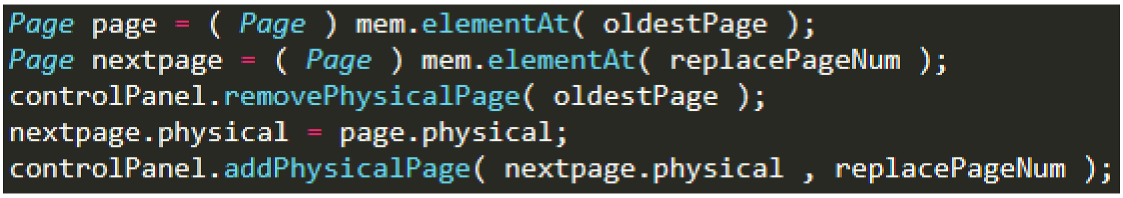
Page number	In memory time (µs)
0	30
1	20
2	10

The oldest page will be the first one.

With modification:

Page number	In memory time (µs)
0	20
1	30
2	10

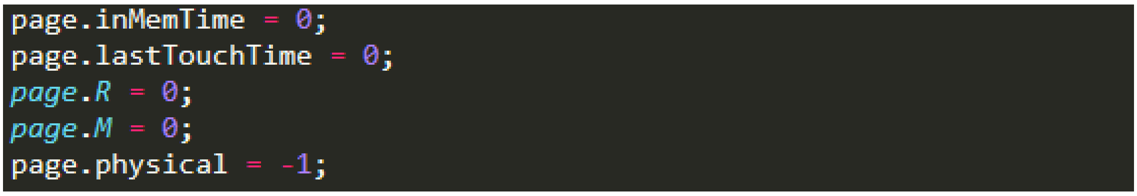
The oldest page will be the second one.



```
Page page = ( Page ) mem.elementAt( oldestPage );
Page nextpage = ( Page ) mem.elementAt( replacePageNum );
controlPanel.removePhysicalPage( oldestPage );
nextpage.physical = page.physical;
controlPanel.addPhysicalPage( nextpage.physical , replacePageNum );
```

The lines above are used to link the oldest page to the page to replace.

- The first line is used to get the information of the oldest page.
- The second line retrieve the information about the page to link.
- Then we remove graphically the oldest physical page.
- After that, we link the two pages.
- To finish, we update the GUI.



```
page.inMemTime = 0;
page.lastTouchTime = 0;
page.R = 0;
page.M = 0;
page.physical = -1;
```

The last lines of this program (above) are used to reset all the information of the previous virtual page.