Paging

Sept 28, 2018

Flaw in Linux

- Meltdown and Spectre CPU flaws
- The kernel bug is a cache invalidation flaw in Linux **memory management** that has been tagged as CVE-2018-17182, reported to Linux kernel maintainers on September 12, 2018

Senior Project Presentation

- Computer Vision project – Quentin Curteman
- Detecting Alexa connected devices – Dylan Cook

Questions

- **When they use terms that are not defined or explained properly**
- **When assumptions are not presented**
- **Using terms that have different meanings**
- **What is the purpose of their project – do you get it?**
- **What is their contribution – did they say it?**
- **What is the status?**

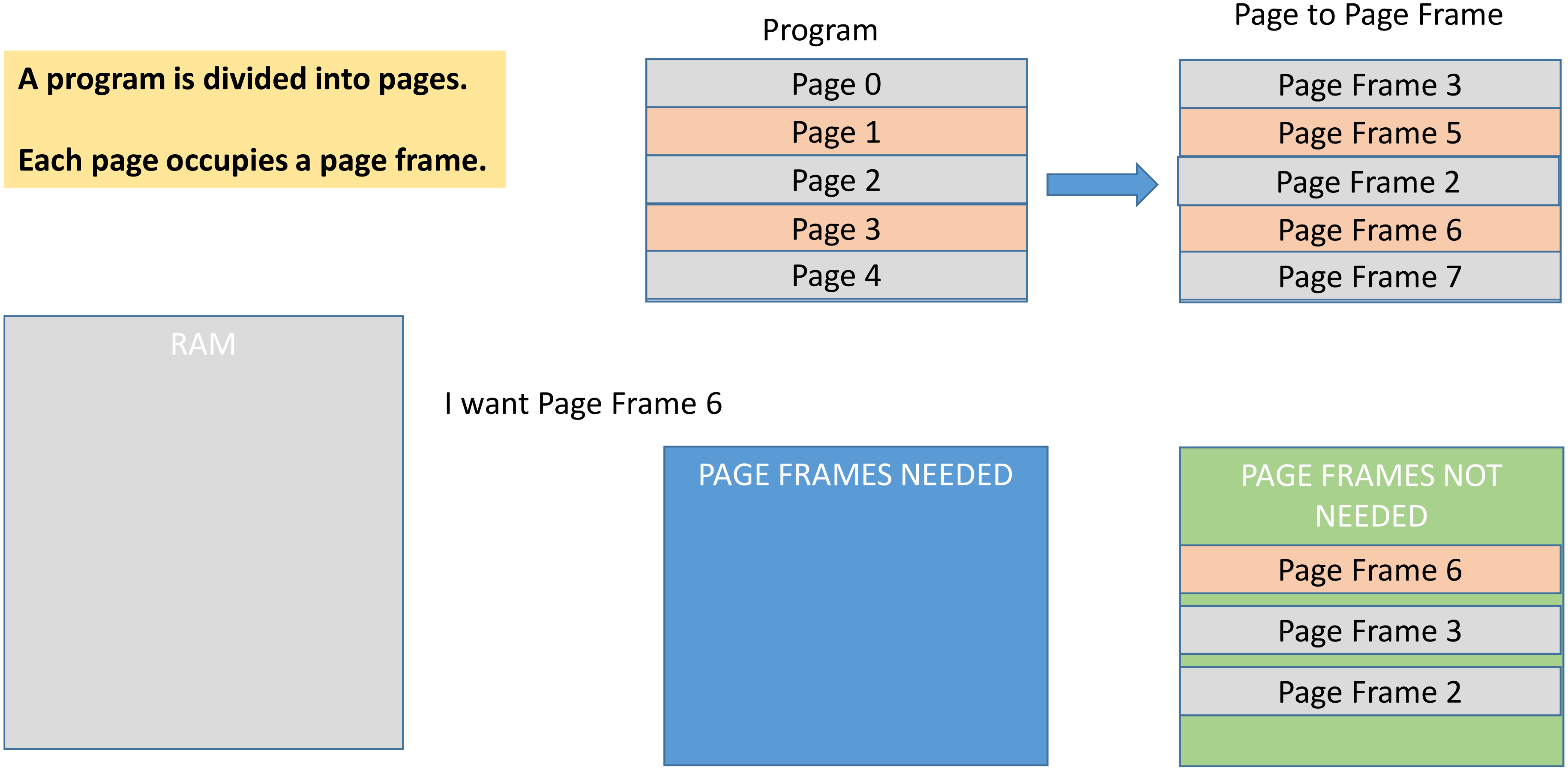
When the process wants a particular page in a file to be loaded into memory

Demand Paging

This is your load_pages()

A program is divided into pages.

Each page occupies a page frame.



Program

Page 0
Page 1
Page 2
Page 3
Page 4

Page to Page Frame

Page Frame 3
Page Frame 5
Page Frame 2
Page Frame 6
Page Frame 7

RAM

I want Page Frame 6

PAGE FRAMES NEEDED

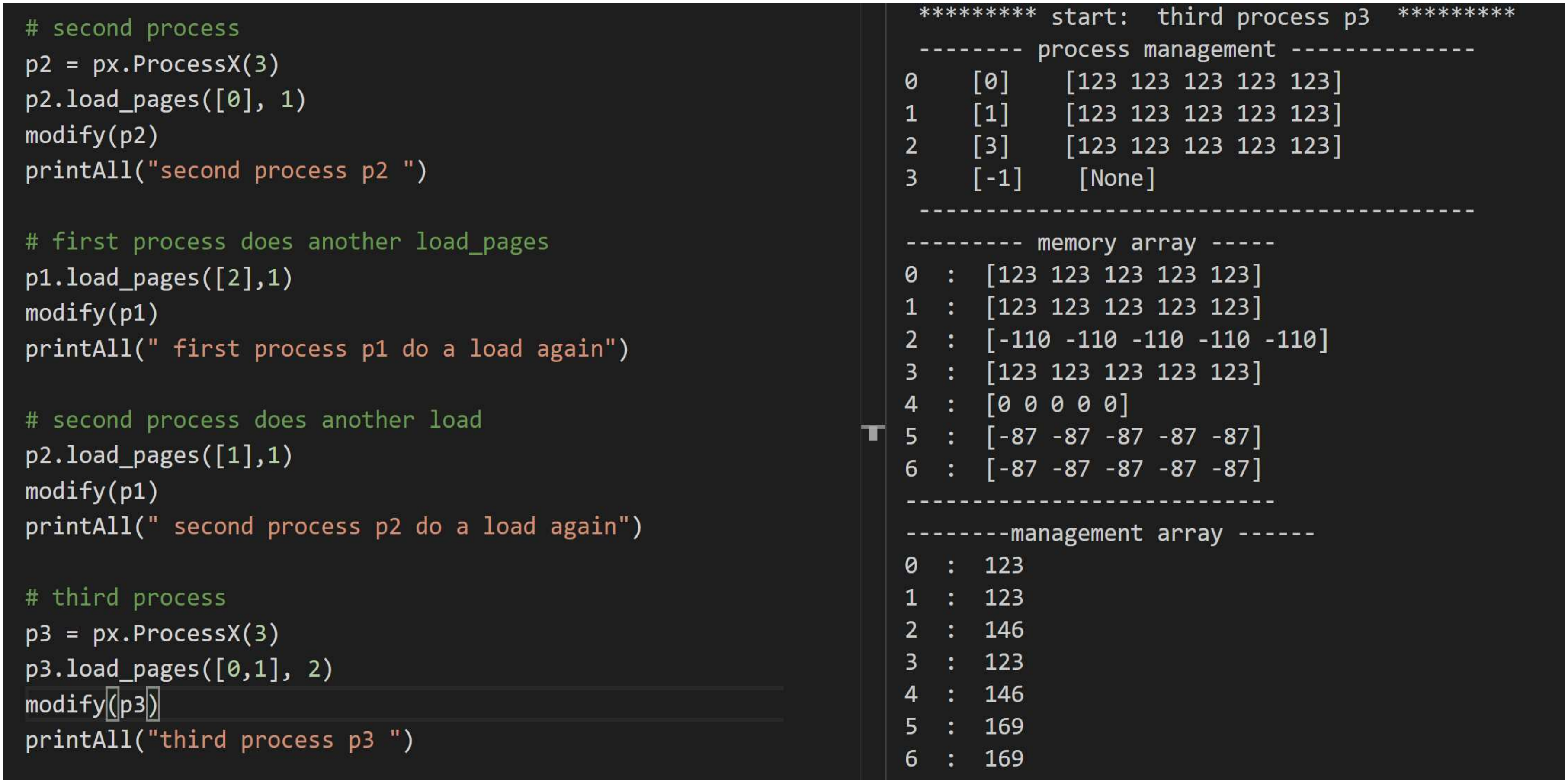
PAGE FRAMES NOT NEEDED

Page Frame 6
Page Frame 3
Page Frame 2

We are ready to tackle the problem

When memory is full

After running test2.py, a 4th process wants 3 page frames



```python
# second process
p2 = px.ProcessX(3)
p2.load_pages([0], 1)
modify(p2)
printAll("second process p2 ")

# first process does another load_pages
p1.load_pages([2],1)
modify(p1)
printAll(" first process p1 do a load again")

# second process does another load
p2.load_pages([1],1)
modify(p1)
printAll(" second process p2 do a load again")

# third process
p3 = px.ProcessX(3)
p3.load_pages([0,1], 2)
modify(p3)
printAll("third process p3 ")
```

```
 *********  start:  third process p3  *********
 -------- process management -------------
0    [0]    [123 123 123 123 123]
1    [1]    [123 123 123 123 123]
2    [3]    [123 123 123 123 123]
3    [-1]     [None]
 ------------------------------------------
--------- memory array -----
0  :  [123 123 123 123 123]
1  :  [123 123 123 123 123]
2  :  [-110 -110 -110 -110 -110]
3  :  [123 123 123 123 123]
4  :  [0 0 0 0 0]
5  :  [-87 -87 -87 -87 -87]
6  :  [-87 -87 -87 -87 -87]
----------------------------
--------management array ------
0  :  123
1  :  123
2  :  146
3  :  123
4  :  146
5  :  169
6  :  169
```

How to determine page replacement policy?

- Memory is not unlimited
- So, eventually we need to remove pages from memory so that new pages may come in

- So, we need a page replacement policy…
  1. First in First Out
  2. Least recently used
  3. Least frequently used
  4. Most recently used

Each policy requires specific algorithms

First In First Out

- Memory page frame that is the oldest is the first to be swapped out
- Question
  - How to keep track of which page frames are the oldest

Policy: First In First Out

- What information do we need for the memory management structures to do this:

Timestamp ??? time.time()

- **Need to compute differences in time**

Counter ??

Any other way? Logical clock?

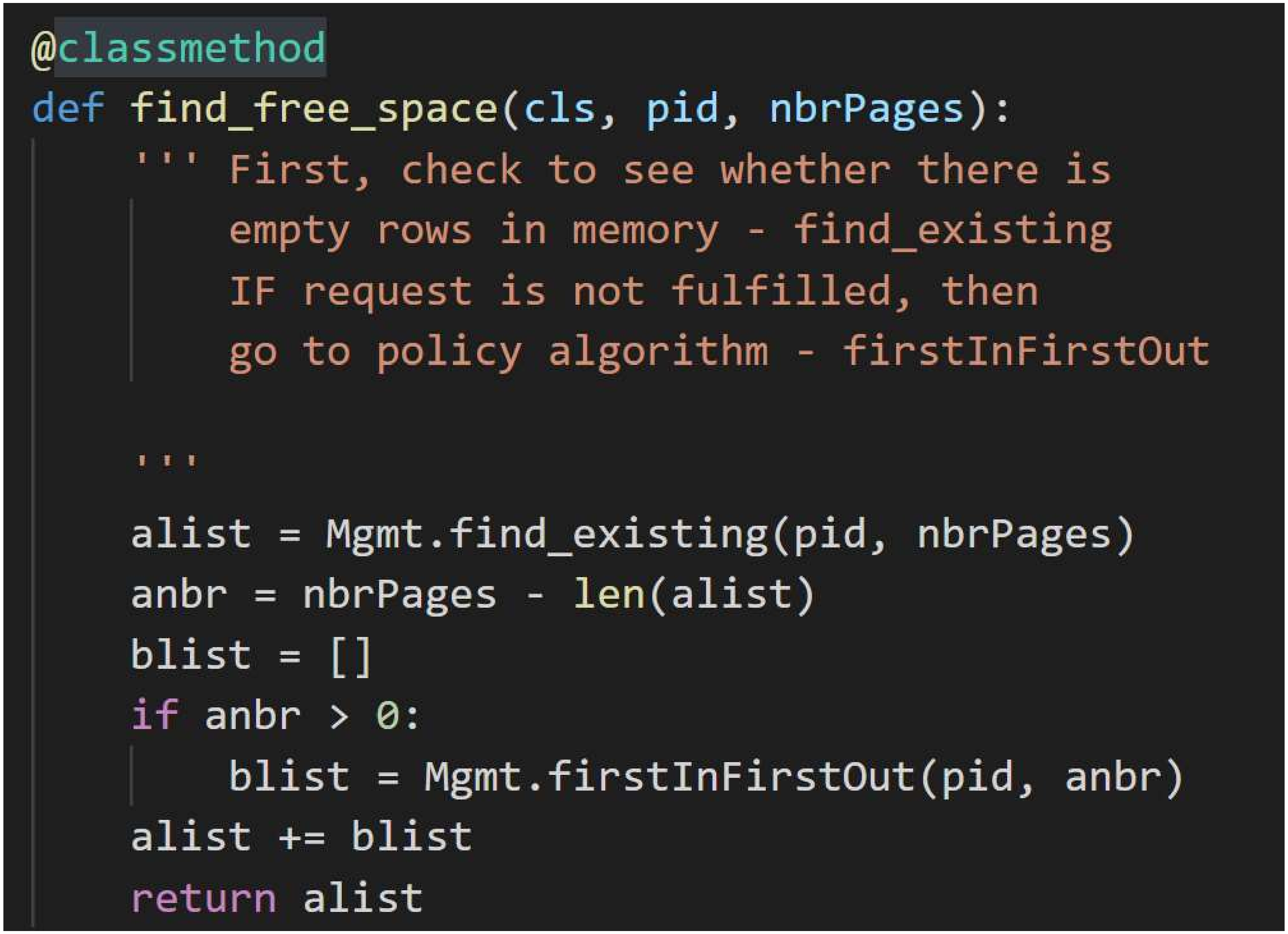
Here are some issues:

- **How to swap out a page Frame?**
- **How to inform process that the particular page Frame is gone?**

Physical Memory Management Structure

Page Frame Index	PID	Age_counter

Keep track when a pid is given a page frame.



```python
@classmethod
def find_free_space(cls, pid, nbrPages):
    ''' First, check to see whether there is
        empty rows in memory - find_existing
        IF request is not fulfilled, then
        go to policy algorithm - firstInFirstOut

    '''
    alist = Mgmt.find_existing(pid, nbrPages)
    anbr = nbrPages - len(alist)
    blist = []
    if anbr > 0:
        blist = Mgmt.firstInFirstOut(pid, anbr)
    alist += blist
    return alist
```

Test2.py

A 4th process wants
3 page frames

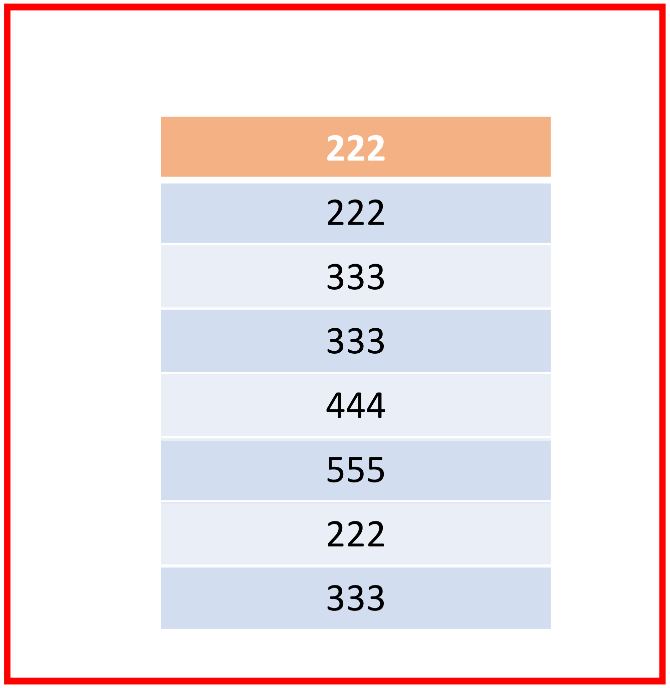
Mgmt

Process
Mgmt

Code Assignment: code-sept-28a
DUE: Oct 1st, 2018

Which page frames will be replaced?

MEMORY



222

222

333

333

444

555

222

333

What new functions and data structures?

- When a new process is created, it need to inform Memory manager its PID and an process object representing the process
  - Most likely to be done in new ProcessX object
    - Telling MemA that a new process is here and need to be registered
    - In MemA
      - Classmethod
      - register_process(cls, object)
        - A new data structure is required to store the pid and object
      - NOTE: the object is used when the page frame occupied by the pid is to be swapped out
        - processObject.removePageFrame(pageFrameIndex)