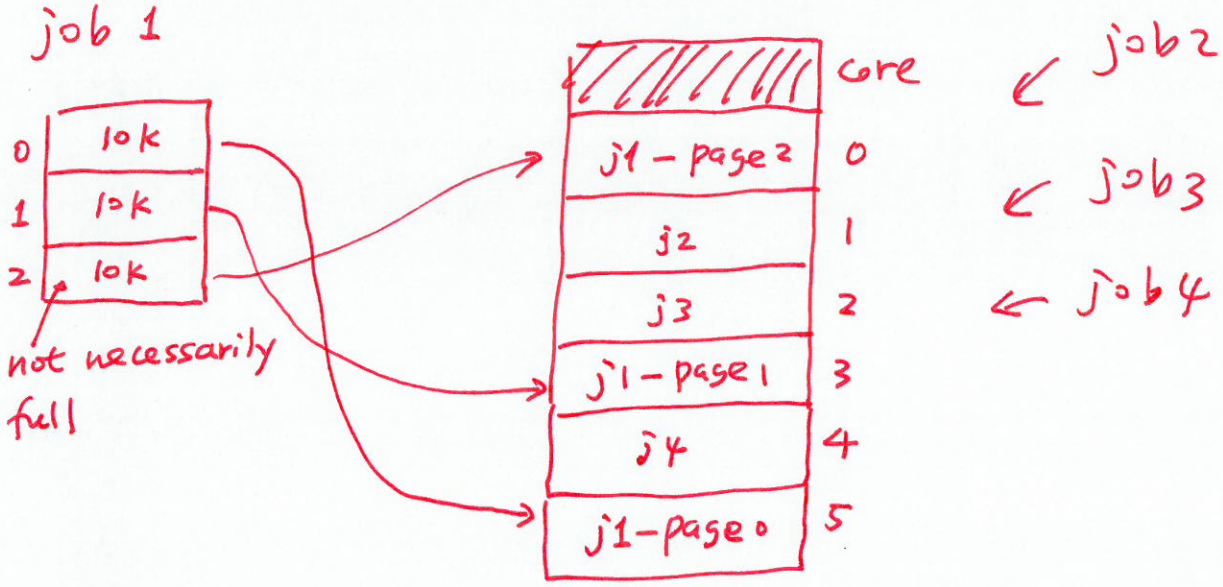
- How do we manage paging?
  - 1. Job Table.
  - 2. Page Map Table (for each job).
  - 3. Memory Map Table.



Job Table

job #	size	PMT location
1	26K	xxxxx

pointer

PMT (for job 1)

page #	page frame number in memory
0	5
1	3
2	0

Memory Map Table

	page frame #	status
0	0	busy
1	1	--
2	2	free
3	3	busy
4	4	--
5	5	busy

if only consider job 1.

- What if we have a goto statement?
- Offset (displacement) of a line is the factor used to locate that line within the page frame.
- Intuitively, offset represents how far away a line is from the beginning of its page.

page #

page size) line # to be located

x x x x

x x x x ← displacement.

Ex. page size – 500 lines.

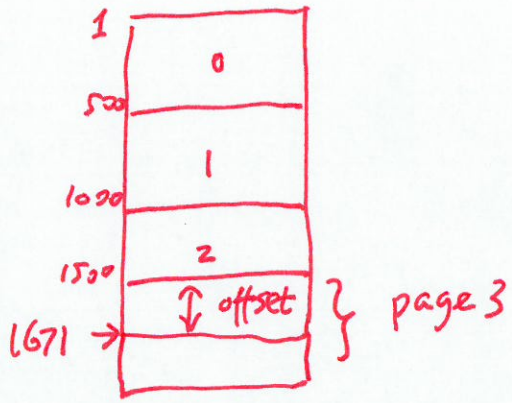
need to access line 1671

3 ← page #

500) 1671

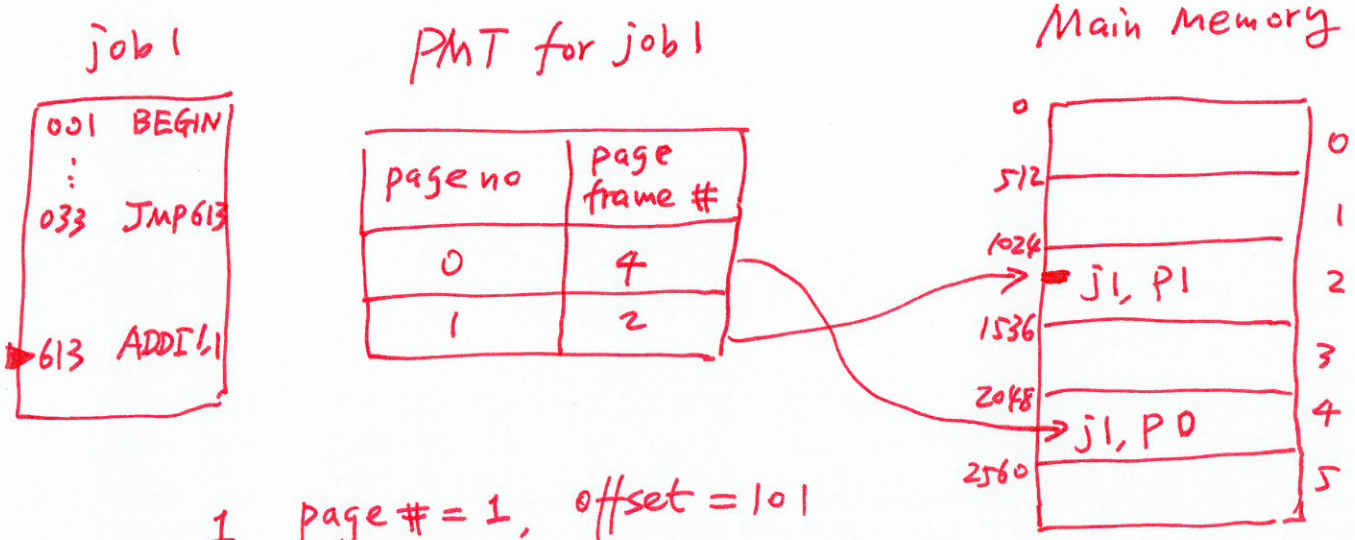
1500

171 ← offset (displacement)



- In general, the following is the method to handle a goto statement (or to access any special line).
  - 1. Using the previous arithmetic computation to compute page # and displacement of the line.
  - 2. Look up this job's PMT to find the page frame which contains this page.
  - 3. page_frame_address = page_frame_num * page_size
  - 4. instruction_address = page_frame_address + displacement.

Ex. page size 512 lines.



1. page # = 1, offset = 101
2. Look up PMT, page frame = 2
3. page_frame_address = 2 * 512 = 1024
3. Instruction_address = 1024 + 101 = 1125

- Advantage of paging.
  - 1. Job is stored non-contiguously in memory.
  - 2. No external fragmentation.
- Disadvantage of paging.
  - 1. Overhead.
  - 2. Internal fragmentation still exists.
  - 3. Page size too small $\rightarrow$ PMT's have large size.
  - 4. Page size too large $\rightarrow$ internal fragmentation increases.

2. Demand Paging

- Demand paging only loads a part of a program into memory for running.
  - 1. Jobs are still decomposed into equally sized pages.
  - 2. Jobs are initially stored in secondary memory.
- Why demand paging is feasible?

Menu-driven system:

1) add a new record
2) delete a record
3) update a record
4) query
⋮

Locality of reference.

- **Demand paging** allows a user to run jobs with less main memory (this is the idea of **virtual memory**: the user would feel that the physical memory is almost infinite, though it is not the case in reality).

- Page Map Table (PMT) needs to be modified.

page #	page frame #	in memory ?	modified ?	referenced recently ?

j1: 30K

0	3	Y		
1	5	Y		
2	0	Y		

j2: 40K

0	4	Y		
1	1	Y		
2		N		
3		N		

j3: 40K

0	6	Y		
1	2	Y		
2		N		
3		N		

Main memory

j1, P2	0
j2, P1	1
j3, P1	2
j1, P0	3
j2, P0	4
j1, P1	5
j3, P0	6

page size: 10K.

Q: What if I need to access j2, P2 ?

- How does the computer fetch an instruction?
  - 1. Start processing instruction
  - 2. Generate data address
  - 3. Compute page number
  - 4. If page is in memory
  - then
  - get data and finish instruction
  - advance to the next instruction
  - return to step 1
  - else
  - generate page interrupt
  - call page interrupt handler