# Phase 2 help

### \*Process Loading:

- Read files one by one. (Each file is for a program)
- Memory is divided into frames of size 128 bytes.
- Each process would have two page tables.
- After writing 128 bytes of the data of the program (i.e. a page) to a frame, assign that frame to that page on the data page table of that process.
- Same goes for code, after writing a page of the code to a frame, assign that frame to that page on the code page table of that process.
- Check code and data size is according to the limit specified.
- After loading a process into memory, create its PCB (i.e. a class), which have all the process's
  info including the reference to the page table you have created for that process. Put the
  reference to that class into a ready queue according to the priority given.
- After loading all the processes, add remaining frames to free page table.

#### =>File Structure:

- 1. 1st byte: priority.
- 2. 2nd & 3rd byte: Process ID (PID)
- 3. 4th & 5th byte: data size
- 4. 6th, 7th, 8th bytes: ignore them
- 5. 8<sup>th</sup> onwards is data and data will be loaded in data segment according to the size given
- 6. After data, code is given in the files

At this point, you have loaded processes, divided memory into frames, assigned pages in page tables and you have processes into queues.

## **CPU Scheduling:**

- There could be a function or class of scheduler which would tell which process to execute next.
- That scheduled process's registers (i.e. stored in PCB) are mapped to actual registers of the processor.
- Move that process (i.e. its PCB's object reference) into running queue (contains only one process).
- Run the process (this you have done in the previous assignment)
  - After its time quantum finishes (if it is 8 cycles, then it becomes 4 instruction), remove that
    process from running queue and store its state (i.e. replaces its PCB's registers with actual
    processor's registers).
  - Update every process's running time and waiting time and call scheduler, which may rearrange the queue if need and give you next process to run.

(Note: \* = updated).

### **Process Termination:**

- After a process terminates, either because of an error or successfully, print its memory dump (its memory contents) and PCB on screen and into a file as well.
- Remove that process from ready queue.
- Add its frames (from its page table) to free page table.
- Reinitialize quantum and schedule another process

# **Memory Management:**

Memory is not contiguous rather divided into segments (pages), therefore when jumping to an instruction, or next instruction pointing to another frame, or accessing data of that process, you need to check whether the corresponding page is of that process or not.

(PTO)

# \*Paging Example:

page size = finme size = 4 bytes  (Index in durind)
(index in decimal) Nemory (divided in farmes) Page Tables:  Flame 0 Process 1  frame 1 (very) Page 0 frame 5  frame 2 Page 1 frame 6  frame 3 Page 1 frame 1  frame 4 Process 2  Frame 5 Page 0 frame 3  page 1 frame 3  page 2 frame 3  page 2 frame 3  page 1 frame 4  page 2 frame 4  page 3 frame 7  page 4 frame 4  page 4 frame 4  page 4 frame 4  page 4 frame 4  page 5 frame 1  page 6 frame 4  page 6 frame 4  page 1 frame 3  page 1 frame 4  page 2 frame 4  page 1 frame 4  page 1 frame 5  page 1 frame 5  page 1 frame 6  page 1 frame 1  page 2 frame 1  page 3 frame 1  page 4 frame 1  page
(index in decimal) Nemory (divided in farmes) Page Tables:  Flame 0 Process 1  frame 1 (very) Page 0 frame 5  frame 2 Page 1 frame 6  frame 3 Page 1 frame 1  frame 4 Process 2  Frame 5 Page 0 frame 3  page 1 frame 3  page 2 frame 3  page 2 frame 3  page 1 frame 4  page 2 frame 4  page 3 frame 7  page 4 frame 4  page 4 frame 4  page 4 frame 4  page 4 frame 4  page 5 frame 1  page 6 frame 4  page 6 frame 4  page 1 frame 3  page 1 frame 4  page 2 frame 4  page 1 frame 4  page 1 frame 5  page 1 frame 5  page 1 frame 6  page 1 frame 1  page 2 frame 1  page 3 frame 1  page 4 frame 1  page
Forest Frame PC/page 5 2 = 13/4 = 3  Consent page = D. Storm process 2 page Table  Consent appset = PC/page 5 2e = 13/4 = 3  Consent page = D. Storm process 2 page Table  Consent appset = PC/page 5 2e = 13/4 = 1.
Forest Frame PC/page 52 = 13/4 = 3  Consent appose = PC 1/2, page 52 = 13/4 = 1.  Instruction = JUNP (0011) -> decimal  Jame 1   June 2   June 3    Process 2   June 3    Process 2   June 4    Process 2   June 5    Process 2   June 6    Process 2   June 7    Process 3   June 7    Process 4   June 7    Process 2   June 7    Proces
frame 1 (key) prop 0 frame 5 (frame 5)  frame 2 prop 1 frame 6  frame 3 prop 2 frame 1  frame 4  frame 5 Process 2  frame 6 prop 0 frame 3  frame 7 prop 2 frame 0  prop 2 frame 1  prop 2 frame 1  prop 2 frame 1  prop 3 frame 7  prop 3 frame 7  prop 3 frame 7  prop 3 frame 1  PC = 13  Consent frame = PC/propesse = 13/4 = 3  consent frame = PC/propesse = 13/4 = 3  consent affect = PC 1/6 propesse = 13 % 4 = 1.  Instruction = JUMP (0011 - decimal)  imm value = 1411  jump propess mm value / 4 = 11/4 = 2
frame 2  frame 3  frame 3  frame 4  frame 5  frame 5  frame 5  frame 6  page 0 frame 3  page 1 frame 1  page 1 frame 1  page 1 frame 0  page 1 frame 0  page 1 frame 0  page 2 frame 1  page 2 frame 1  page 3 frame 1  PC = 13  Consent frame = PC/page 52e = 13/4 = 3  consent frame = PC/page 52e = 13/4 = 3  consent apper = D. fram process 2 page Table  consent apper = PC 1/2 page 52e = 13/4 = 1.  Instruction = JUMP (0011 - decoral  imm value = 1111  jump pages = 1111
frame 4  Frame 5  Pocess 2  Frame 6  Page D Frame 3  Frame 7  Page 1 Harne 0  Page 2 Harne 1  Page 3 Harne 1  Pree Page Table  Consent frame = PC/page 42e = 13/4 = 3  Consent page = 0. Fram process 2 page Table  Consent application = JUMP (0011 - decimal)  Instruction = JUMP (0011 - decimal)  Jump pages man value / 4 = 11/4 = 2
For earple,  Consent frame = PC/page 42e = 13/4 = 3  Consent page = D. (from process 2 page Table)  Consent apport = PC %, page 52e = 13% 4 = 1.  Instruction = JUMP (0011 -> decimal)  Jump pages mm value/4 = 11/4 = 2
For earple,  PC = 13  Cutsent frame = PC/page 5:2e = 13/4 = 3  cutsent page = D. (from process 2 page Table)  cutent 4/65ct = PC % page 5:2e = 13% 4 = 1.  Instruction = JUNIP (0011 -> decimal imm value = 1411  jump pages= mm value/4 = 11/4 = 2
For examples  For examples  Page 1 frame 0  Page 2 frame 1  Page 3 frame 1  Page 3 frame 1  Page 3 frame 4  PC = 13  Consent frame = PC/page 52e = 13/4 = 3  Consent page = 0. (from process 2 page Table)  Consent aggset = PC 1/6 page 52e = 13 1/4 = 1.  Instruction = JUMP (0011 -> decimal)  imm value = 1411  jump pages = mm value /4 = 11/4 = 2
For example,  Vunning queue = [psacess 2]  PC = 13  Consent frame = PC/page 52e = 13/4 = 3  consent page = 0. (from process 2 page Table)  consent qqqset = PC 1/2 page 52e = 13 1/4 = 1.  Instruction = JUMP (0011 -> decimal)  jump pages= mm value/4 = 11/4 = 2
Page 3   frame 7  Whining queue = [psacess 2] Free Page Table  PC = 13  Consent frame = PC/page 52e = 13/4 = 3  Consent page = 0. [from process 2 page Table]  Consent appear = PC 1/6 page 52e = 13 1/6 4 = 1.  Instruction = JUMP (0011 -> decimal)  Imm value = 11/1  jump pages = 11/4 = 2
winning queue = [psacess ]  PC = 13  Consent frame = PC/page size = 13/4 = 3  consent page = 0. from process 2 page Table)  consent appear = PC 1/6 page size = 13 % 4 = 1.  Instruction = JUMP (0011 -> decimal)  imm value = 11/1  jump pages = 11/4 = 2
PC = 13  Consent frame = PC/page 52e = 13/4 = 3  consent appet = D. (from process 2 page Table)  consent appet = PC % page 52e = 13% 4 = 1.  Instruction = JUMP (0011 - heimal  imm value = #11  jump pages = mm value/4 = 11/4 = 2
consent frame = PC/page 52e = 13/4 = 3  consent page = 0. (from process 2 page Table)  consent 4/65ct = PC 1/6 page 52e = 13 1/6 4 = 1.  Instruction = JUMP (0011 -> decimal  imm value = 11/1  jump pages = 11/4 = 2
consent frame = PC/page 52e = 13/4 = 3  consent page = 0. (from process 2 page Table)  consent 4/65ct = PC 1/6 page 52e = 13 1/6 4 = 1.  Instruction = JUMP (0011 -> decimal)  imm value = 1111  jump pages = 11/4 = 2
consent page = 0. (from process 2 page Table)  consent appoint = PC % page size = 13% 4 = 1.  Instruction = JUMP (0011 -> decimal  imm value = 1111  jump pages = man value/4 = 11/4 = 2
consent page = 0. (from process 2 page Table)  consent appoint = PC % page size = 13% 4 = 1.  Instruction = JUMP (0011 -> decimal  imm value = 1111  jump pages = man value/4 = 11/4 = 2
Instruction = JUMP (0011 - decimal)  Jump pages = mm value/4 = 11/4 = 2
Instruction = JUMP (0011 - decimal)  mm value = 411  jump pages= mm value/4 = 11/4 = 2
Instruction = JUMP (0011 - decimal)  mm value = 411  jump pages= mm value/4 = 11/4 = 2
jump pages = mm value/4 = 11/4 = 2
jump pages = mm value/4 = 11/4 = 2
Jump pages= 1 1/4 = 2 Jump aggsets = 1 mm value 1/6 4 = 11 1/6 4 = 3
1 mp aggreets = 1 mm value 1/6 9 = 11 1/6 4 = 3
1 1 to the country of the
(combined ) agreet = convent agreet + jump agreets = 1+3=4

	Pated:	
	is aggreet >, page size (4) {	
	jump pages = jump pages +1 = 2+1=)3  aggset = aggset % pagesize (4) = 474 => 0	
	ageset = ageset % pagesize (4) = 1873	
7	page No (page to jump) = correct page + jump pages = 0 + 3 => 3	
→ →		
y yes	, frame No. (frame to j-mp) = 7 (from page table of process?	
- dre	(unos)	
-	(value of) PC = (frame No. x frame size) + affect	
7	PC = (7 ×4) + 0 => 28	
3		
9		