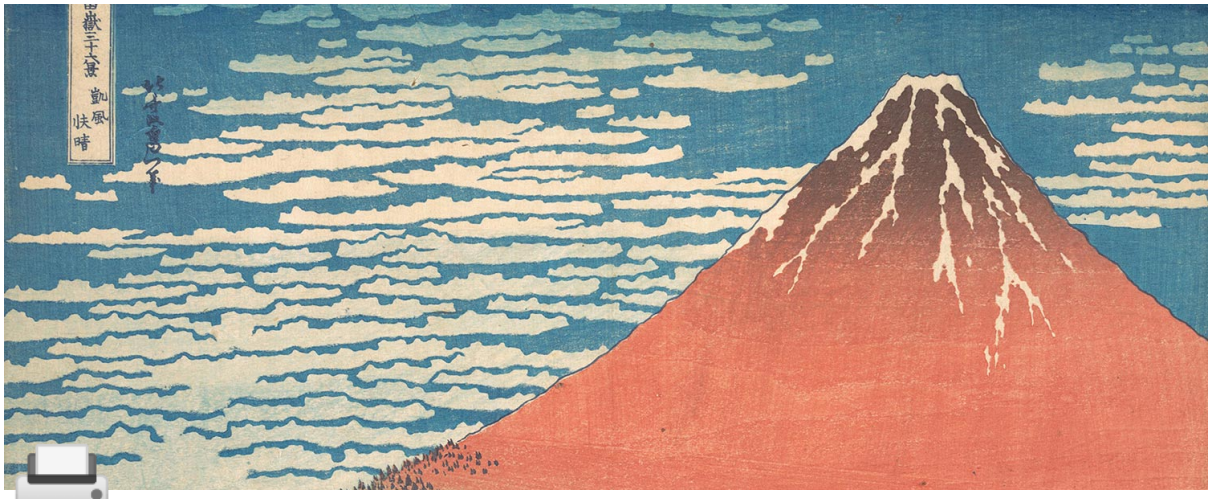


popquiz8

CS5250 Advanced Operating Systems

Pop Quiz 8

Name: _____Ding Fan_____

Student Number: __A0248373X_____

Suppose there are three tasks, A, B, and C, of niceness levels -2, 0, 2, respectively. Assume that they arrived at the same time. Show the first 10 context switches that will happen under CFS and BFS. Please show your working, i.e., the intermediate state of the corresponding data structures. State any assumption clearly.

CFS

solution:

- the task A B C arrived the same time.
- Assume `min_vruntime = 0`
- Assume `period = 6ms`
- because A, B, and C, of niceness levels -2, 0, 2. the load weight of `A, B, and C is 1586, 1024, 655`
- Time slice for a task

$$TimeSliceForaTask = period * (weightOfTask)/(totalWeightOfRunqueue)$$

- When the scheduler is called, the current task' vruntime is updated:

  $$vruntime+ = (timeprocessran) * 1024/(loadweightofthisprocess)$$

beginning and Context switch1

- the task A B C arrived the same time. the have the same `vruntime = min_vruntime=0`
- assume A wake first
- Time slice for task A

  $$TimeSliceFortaskA = period * (weightOfTask)/(totalWeightOfRunqueue) = 6 * 1586/(1586 + 1024 + 655) = 2.914548$$

- task A take over the cpu, it run for 2.914548ms
- recompute the `vruntime` of A, and add A to the Red Black tree

  $$vruntimeA+ = (timeProcessRan) * 1024/(loadWeightOfThisProcess) = 2.914548 * 1024/1586 = 1.881776$$

Context switch2

- `vruntimeB = 0`, `vruntimeC = 0`, `vruntimeA=1.881776`
- BC has the same minimal `vruntime`, assume B wake first
- Time slice for task B

  $$TimeSliceFortaskB = period * (weightOfTask)/(totalWeightOfRunqueue) = 6 * 1024/(1586 + 1024 + 655) = 1.881776$$

- task B take over the cpu, it run for 1.881776ms
- recompute the `vruntime` of B, and add B to the Red Black tree

  $$vruntimeB+ = (timeProcessRan) * 1024/(loadWeightOfThisProcess) = 1.881776 * 1024/1024 = 1.881776$$

Context switch3

- `vruntimeC = 0`, `vruntimeA = 1.881776`, `vruntimeB=1.881776`
- C has the minimal `vruntime`, C wake
- Time slice for task C

  $$TimeSliceFortaskC = period * (weightOfTask)/(totalWeightOfRunqueue) = 6 * 655/(1586 + 1024 + 655) = 1.203368$$

- task C take over the cpu, it run for 1.203368ms

- recompute the `vruntime` of C, and add C to the Red Black tree

  $vruntimeB+=(timeProcessRan)*1024/(loadWeightOfThisProcess)=1.203368*1024/655=1.881776$

Context switch4

- `vruntimeA = 1.881776`, `vruntimeB = 1.881776`, `vruntimeC=1.881776`
- the have the same `vruntime`, assume A wake first
- Time slice for task A

  $TimeSliceFortaskA=period*(weightOfTask)/(totalWeightOfRunqueue)=6*1586/(1586+1024+655)=2.914548$

- task A take over the cpu, it run for 2.914548ms
- recompute the `vruntime` of A, and add A to the Red Black tree

  $vruntimeA+=(timeProcessRan)*1024/(loadWeightOfThisProcess)=1.881776+2.914548*1024/1586=3.763552$

Context switch5

- `vruntimeB = 1.881776`, `vruntimeC = 1.881776`, `vruntimeA=3.763552`
- BC has the same minimal `vruntime`, assume B wake first
- Time slice for task B

  $TimeSliceFortaskB=period*(weightOfTask)/(totalWeightOfRunqueue)=6*1024/(1586+1024+655)=1.881776$

- task B take over the cpu, it run for 1.881776ms
- recompute the `vruntime` of B, and add B to the Red Black tree

  $vruntimeB+=(timeProcessRan)*1024/(loadWeightOfThisProcess)=1.881776+1.881776*1024/1024=3.763552$

Context switch6

- `vruntimeC = 1.881776`, `vruntimeA = 3.763552`, `vruntimeB=3.763552`
- C has the minimal `vruntime`, C wake
- Time slice for task C

  $TimeSliceFortaskC=period*(weightOfTask)/(totalWeightOfRunqueue)=6*655/(1586+1024+655)=1.203368$

- task C take over the cpu, it run for 1.203368ms
- recompute the `vruntime` of C, and add C to the Red Black tree

$$vruntimeB+=(timeProcessRan)*1024/(loadWeightOfThisProcess)=1.881776+1.203368*1024/655=3.763552$$

Context switch7

- `vruntimeA = 3.763552`, `vruntimeB = 3.763552`, `vruntimeC=3.763552`
- the have the same `vruntime`, assume A wake first
- Time slice for task A

  $$TimeSliceFortaskA=period*(weightOfTask)/(totalWeightOfRunqueue)=6*1586/(1586+1024+655)=2.914548$$
- task A take over the cpu, it run for 2.914548ms
- recompute the `vruntime` of A, and add A to the Red Black tree

  $$vruntimeA+=(timeProcessRan)*1024/(loadWeightOfThisProcess)=3.763552+2.914548*1024/1586=5.645328$$

Context switch8

- `vruntimeB = 3.763552`, `vruntimeC = 3.763552`, `vruntimeA=4.96692`
- BC has the same minimal `vruntime`, assume B wake first
- Time slice for task B

  $$TimeSliceFortaskB=period*(weightOfTask)/(totalWeightOfRunqueue)=6*1024/(1586+1024+655)=1.881776$$
- task B take over the cpu, it run for 1.881776ms
- recompute the `vruntime` of B, and add B to the Red Black tree

  $$vruntimeB+=(timeProcessRan)*1024/(loadWeightOfThisProcess)=3.763552+1.881776*1024/1024=5.645328$$

Context switch9

- `vruntimeC = 1.881776`, `vruntimeA = 4.96692`, `vruntimeB=4.96692`
- C has the minimal `vruntime`, C wake
- Time slice for task C

  $$TimeSliceFortaskC=period*(weightOfTask)/(totalWeightOfRunqueue)=6*655/(1586+1024+655)=1.203368$$
- task C take over the cpu, it run for 1.203368ms
- recompute the `vruntime` of C, and add C to the Red Black tree

$$vruntimeB+=(timeProcessRan)*1024/(loadWeightOfThisProcess)=3.763552+1.203368*1024/655=5.645328$$

Context switch10

- `vruntimeA = 4.96692`, `vruntimeB = 4.96692`, `vruntimeC=4.96692`
- the have the same `vruntime`, assume A wake first
- Time slice for task A

  $$TimeSliceFortaskA=period*(weightOfTask)/(totalWeightOfRunqueue)=6*1586/(1586+1024+655)=2.914548$$
- task A take over the cpu, it run for 2.914548ms
- recompute the `vruntime` of A, and add A to the Red Black tree

  $$vruntimeA+=(timeProcessRan)*1024/(loadWeightOfThisProcess)=5.645328+2.914548*1024/1586=7.527104$$

In conclusion the run sequence is A B C A B C A B C A

BFS

solution：

$$Virtualdeadline=niffies+priorityratios[priority]*rrInterval*scalingFactor$$

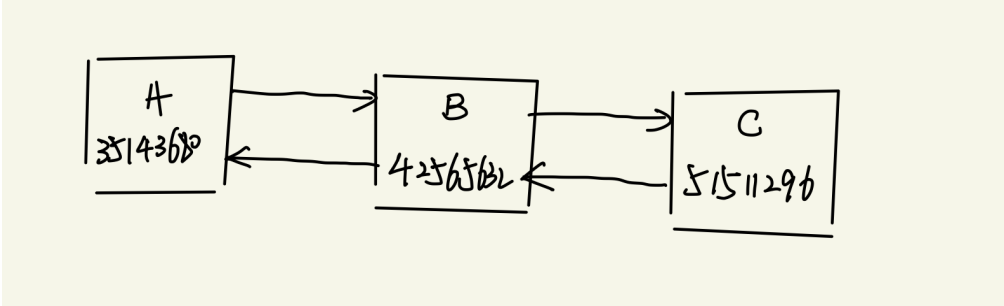
- the task A B C arrived the same time.
- Assume the current niffies time is 0
- Assume the `rrInterval` is 6
- Assume the `scalingFactor` is 8192
- after looking up the value of `priorityratios[priority]`, `priorityratios[-2]`=715, `priorityratios[0]`=866, `priorityratios[2]`=1048
- assume the task A B C are normal task, they have the same priority 102nd
- the run queue is a double linked list

beginning and Context switch1

- $VirtualdeadlineA=0+715*6*8192=35143680$

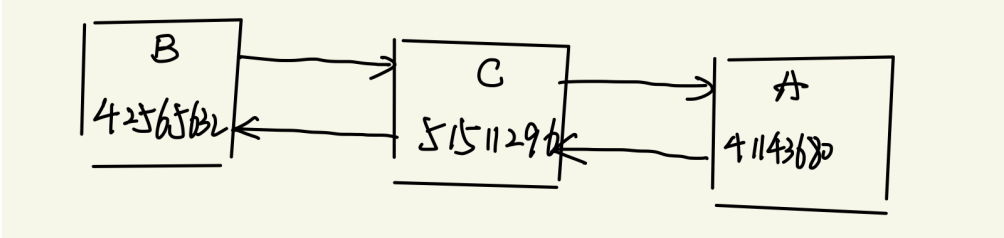
  $VirtualdeadlineB=0+866*6*8192=42565632$

  $VirtualdeadlineC=0+1048*6*8192=51511296$



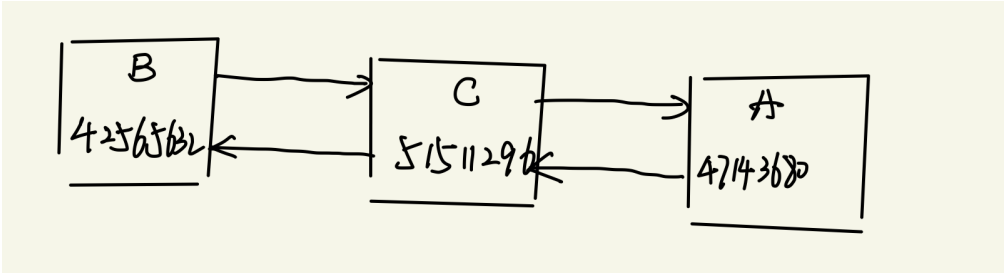
- currently the task A in the queue has the smallest virtual deadline
- task A take over the cpu, it run for 6ms (6000000 nanaseconds)
- recompute and add A to the queue

  $Virtual deadline A = 6000000 + 715 * 6 * 8192 = 41143680$



- currently the task A in the queue has the smallest virtual deadline
- task A is already in CPU do not need context switch, it continue run for 6ms (6000000 nanaseconds)
- recompute and add A to the queue

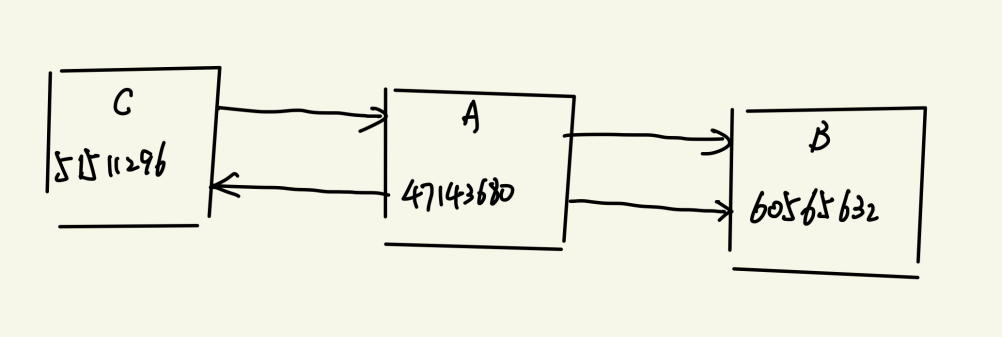
  $Virtual deadline A = 6000000 + 715 * 6 * 8192 = 47143680$



Context switch2

- currently the task B in the queue has the smallest virtual deadline
- task B take over the cpu, it run for 6ms (6000000 nanaseconds)
- recompute and add B to the queue

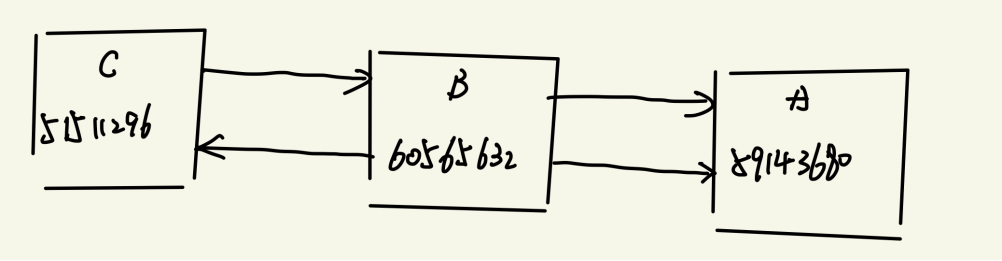
  $Virtual deadline B = 18000000 + 866 * 6 * 8192 = 60565632$



Context switch3

- currently the task A in the queue has the smallest virtual deadline
- task A take over the cpu, it run for 6ms (6000000 nanaseconds)
- recompute and add A to the queue

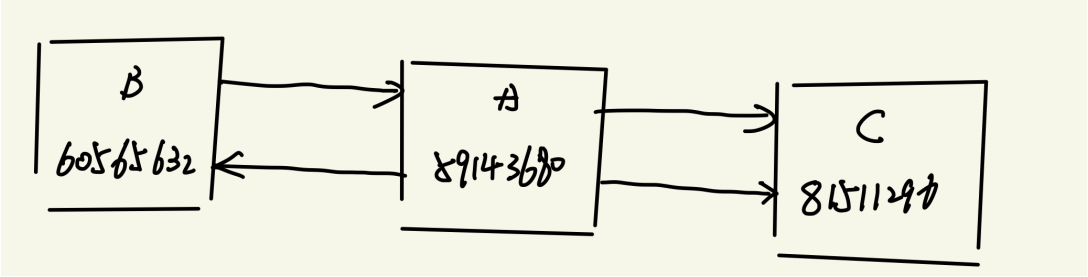
  $Virtual deadline A = 24000000 + 715 * 6 * 8192 = 59143680$



Context switch4

- currently the task C in the queue has the smallest virtual deadline
- task C take over the cpu, it run for 6ms (6000000 nanaseconds)
- recompute and add C to the queue

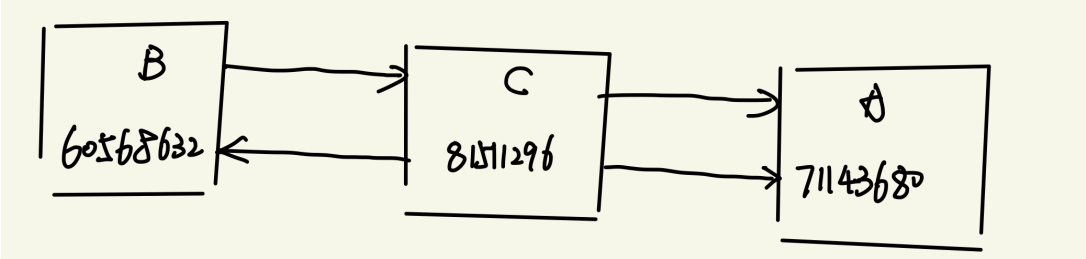
  $Virtual deadline C = 30000000 + 1048 * 6 * 8192 = 81511296$



Context switch5

- currently the task A in the queue has the smallest virtual deadline
- task A take over the cpu, it run for 6ms (6000000 nanaseconds)
- recompute and add A to the queue

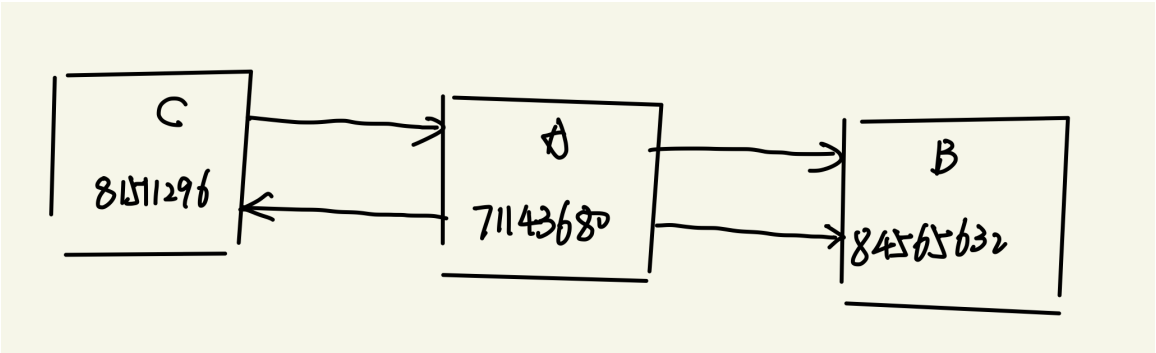
$$Virtual deadline A = 36000000 + 715 * 6 * 8192 = 71143680$$



Context switch6

- currently the task B in the queue has the smallest virtual deadline
- task B take over the cpu, it run for 6ms (6000000 nanaseconds)
- recompute and add B to the queue

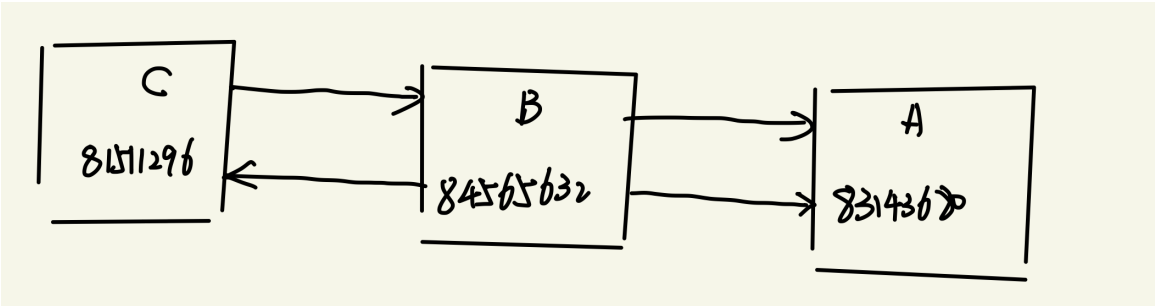
  $$Virtual deadline B = 42000000 + 866 * 6 * 8192 = 84565632$$



Context switch7

- currently the task A in the queue has the smallest virtual deadline
- task A take over the cpu, it run for 6ms (6000000 nanaseconds)
- recompute and add A to the queue

  $$Virtual deadline A = 48000000 + 715 * 6 * 8192 = 83143680$$

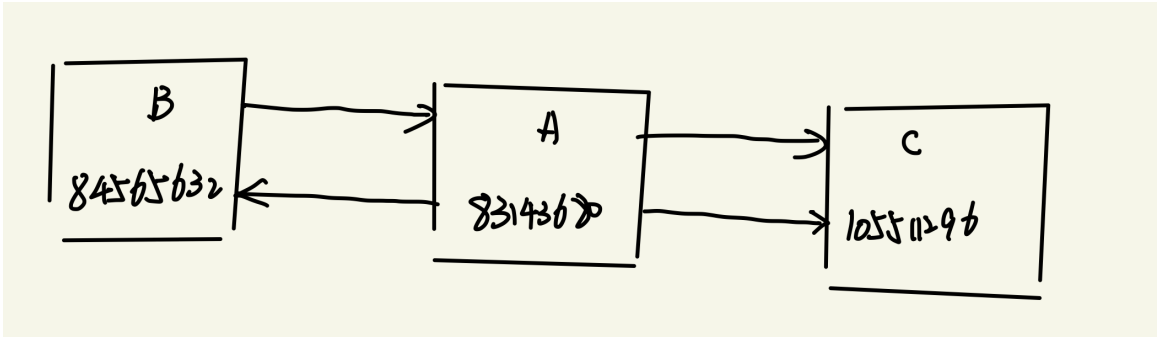


Context switch8

- currently the task C in the queue has the smallest virtual deadline

- task C take over the cpu, it run for 6ms (6000000 nanaseconds)
- recompute and add C to the queue

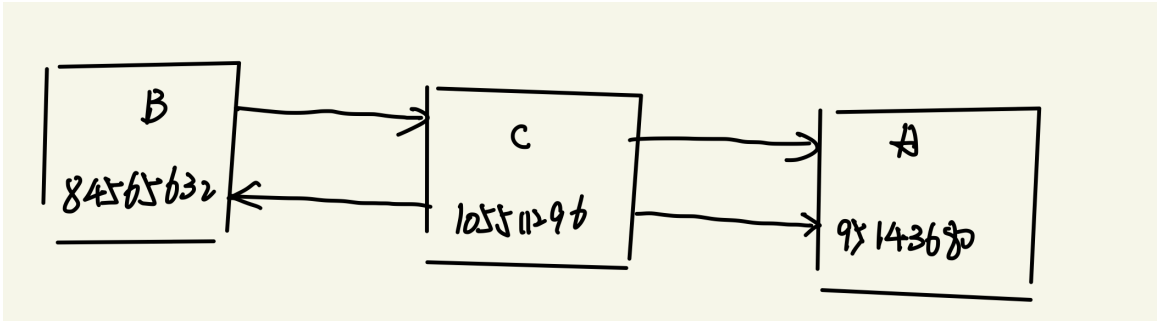
  $Virtual deadline C = 54000000 + 1048 * 6 * 8192 = 105511296$



Context switch9

- currently the task A in the queue has the smallest virtual deadline
- task A take over the cpu, it run for 6ms (6000000 nanaseconds)
- recompute and add A to the queue

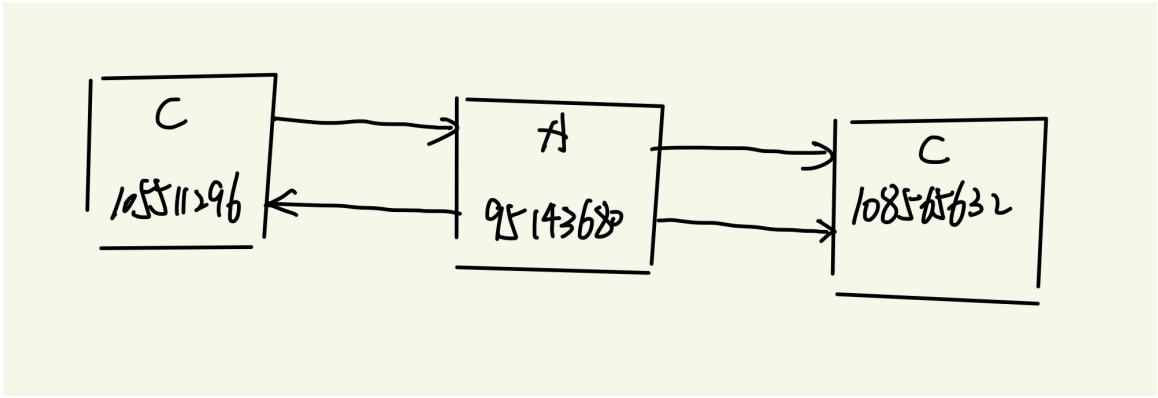
  $Virtual deadline A = 60000000 + 715 * 6 * 8192 = 95143680$



Context switch10

- currently the task B in the queue has the smallest virtual deadline
- task B take over the cpu, it run for 6ms (6000000 nanaseconds)
- recompute and add B to the queue

  $Virtual deadline B = 66000000 + 866 * 6 * 8192 = 108565632$



In conclusion the run sequence is AA B A C A B A C A B