IN1011

Operating Systems

Tutorial 3:

CPU Scheduling…Illustrated

Instructions

- Using a simple scenario with 4 processes, these slides illustrate the consequences of two of the scheduling policies we discussed in this week's lecture – **FCFS** and **Round Robin**;
- The slides also illustrate how to compute a number of important performance measures. These are useful when evaluating the effects of employing different scheduling policies;
- Go through these illustrations carefully, understanding how the policy determines the order in which the processes are assigned to the CPU. Once you have understood the pictures, do the following two tasks:
  - Using the same processes in the illustrated examples, draw similar illustrations for the **Shortest Process Next** (SPN), **Shortest Remaining Time** (SRT) and **Round Robin** policies. To get you started there are already two of the Round Robin illustrations in the slides;
  - Compute throughput, average turnaround time, average wait time and average response time in these cases;

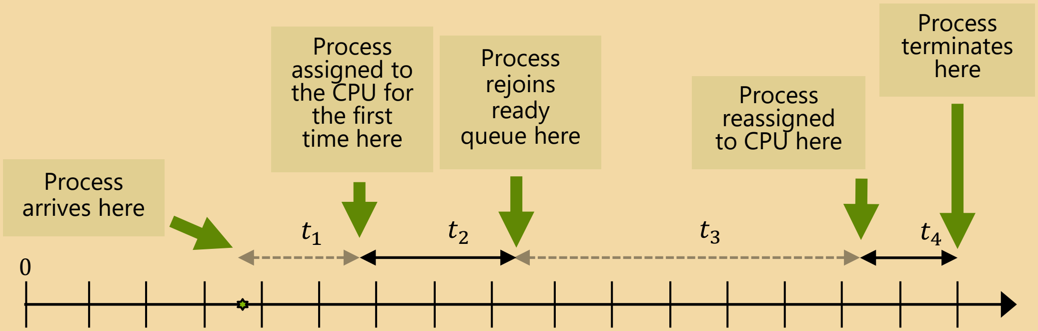
Quantitative Performance Measures

- **User oriented**
  - **Response Time**: time between when the process *arrives in the ready queue* and when it is assigned to the CPU for the first time**;
  - **Turnaround Time**: time between when the process *arrives in the ready queue* and when it finishes executing;
- **System oriented**
  - **Throughput**: is the number of processes that start and complete during the *duration of observing the system*.
    - The duration of observing the system is the time between when the first process arrives and when the last process terminates.
  - **Wait Time**: for each process, the total time the process spends waiting to be assigned to the CPU. Includes the *response time*;
- These definitions assume no overhead in the creation of a process, and negligible time taken to admit a new process to the ready queue;

**sometimes response time is defined from when the process is created, rather than from when the process first enters the ready queue. All the calculations in IN1011 will be from when the process enters the ready queue

Quantitative performance Measures

Response time = t_1
Wait time = Response time $+ t_3$
CPU burst time = $t_2 + t_4$
Turnaround time = CPU burst time + Wait time

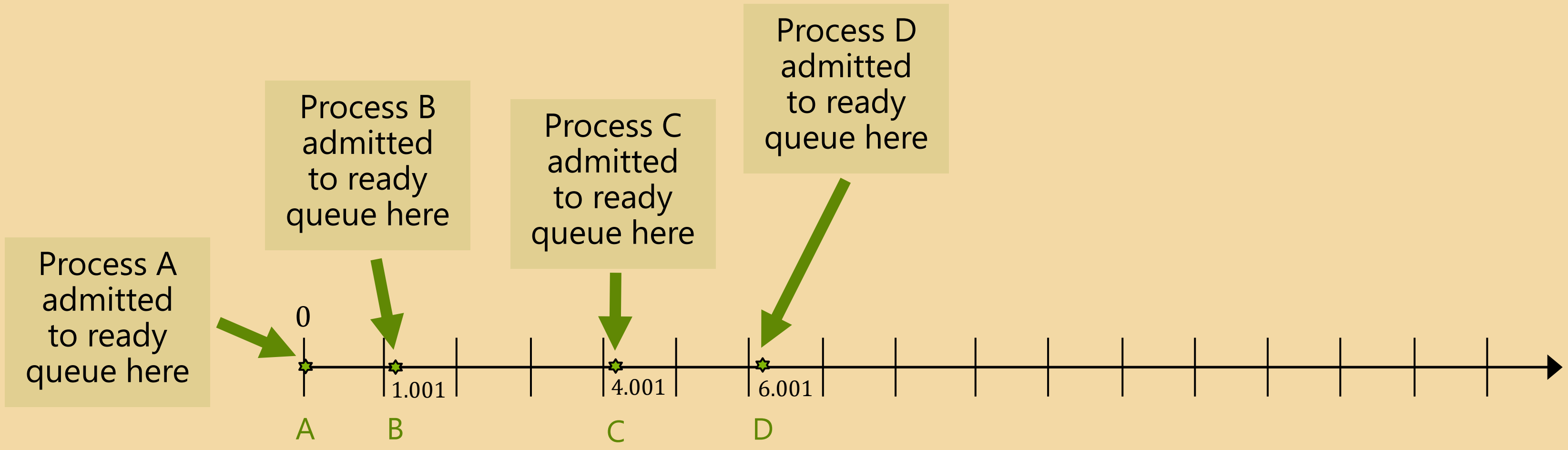


Process arrives here
Process assigned to the CPU for the first time here
Process rejoins ready queue here
Process reassigned to CPU here
Process terminates here
t_1
t_2
t_3
t_4
0

Arrival And Service Times

- Processes A, B, C and D are admitted into the ready queue (by the **long term scheduler**) at times 0, 1.001, 4.001 and 6.001;
- Their respective estimated service times (s in the lecture slides) – that is, how long they need the CPU for – are 3, 6, 4, 2 units of time;
- Service times are also called CPU burst times;
- Let's make some simplifying assumptions:
  - Assume all of these processes are CPU bound, with no I/O requests;
  - Assume all of these processes will successfully complete their executions;
  - Assume no overhead in either assigning processes to the CPU, or freeing the CPU from processes;
  - Assume no other processes;
  - Assume processes always join the ready queue when they are created
- Using a **first-come first-served** (FCFS) scheduling policy, we now illustrate the order in which these processes are assigned to the CPU (by the **short term scheduler** and **dispatcher**). And how long they execute for, when they are assigned;
- We compute the average response time, average turnaround time, throughput and average wait time resulting from the FCFS policy;

Arrival Times



Process A admitted to ready queue here

Process B admitted to ready queue here

Process C admitted to ready queue here

Process D admitted to ready queue here

First-Come First-Served (FCFS)

- Selection function: under the FCFS policy and the assumptions on slide 3, the steps are:
  - When a process is assigned to the CPU, it runs till it terminates;
  - When the currently running process terminates, the process that has been in the ready queue the longest is selected for running;
- Decision mode: non-preemptive – so, with the assumptions in this example, a process will run on the CPU till completion;

FCFS

1) Under FCFS, process A is assigned to the CPU first and runs until completion (remember, slide 3 says this will take 3 units of time)

2) Process B arrives in the ready queue while process A is running

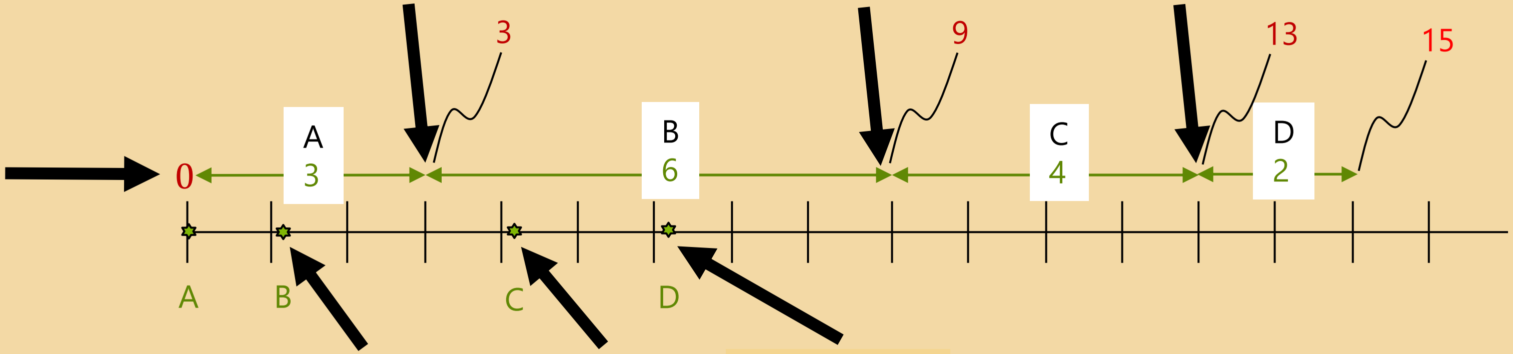
3) Then, at time 3, process A completes and process B is assigned to the CPU (since it was the second to arrive), running to completion in 6 units of time

4) Process C arrives in the ready queue while process B is running

5) Process D joins process C in the ready queue, while process B is still running

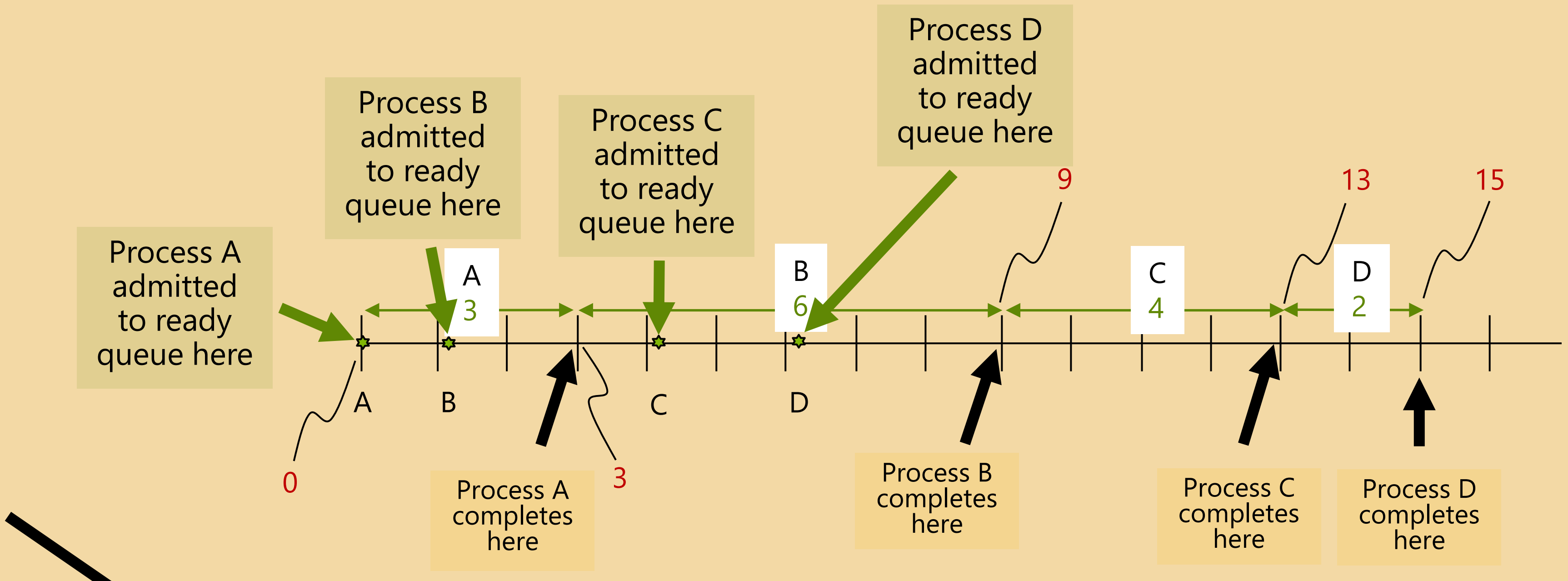
6) Then, at time 9, process B completes and process C is assigned to the CPU (since it was the third to arrive), running to completion in 4 units of time

7) **F**inally, at time 13, process D is assigned to the CPU (since it was the fourth to arrive), running to completion in 2 units of time



FCFS : Throughput

<u>Throughput</u> is the number of processes that start and complete during the *duration of observing the system*. The duration of observing the system is the time between when the first process arrives in the ready queue and when the last process terminates.

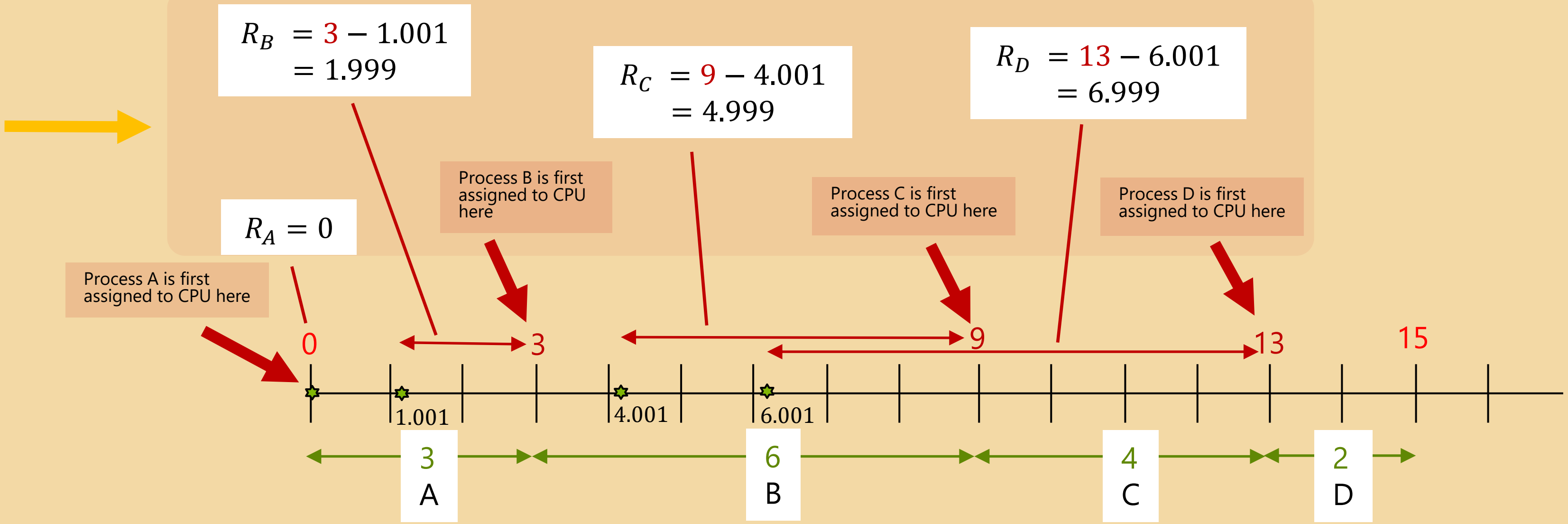


Process A admitted to ready queue here
Process B admitted to ready queue here
Process C admitted to ready queue here
Process D admitted to ready queue here
A
3
B
6
C
4
D
2
0
3
9
13
15
A
B
C
D
Process A completes here
Process B completes here
Process C completes here
Process D completes here

$$\text{Throughput} = \frac{number\ of\ executed\ processes}{duration\ of\ observing\ system} = \frac{4}{15} = 0.267 \text{ processes per unit of time}$$

FCFS: Response Time

Response times for each of the processes (denoted R_i for process i). This is the length of time between when a process enters the ready queue and when it is assigned to the CPU for the first time

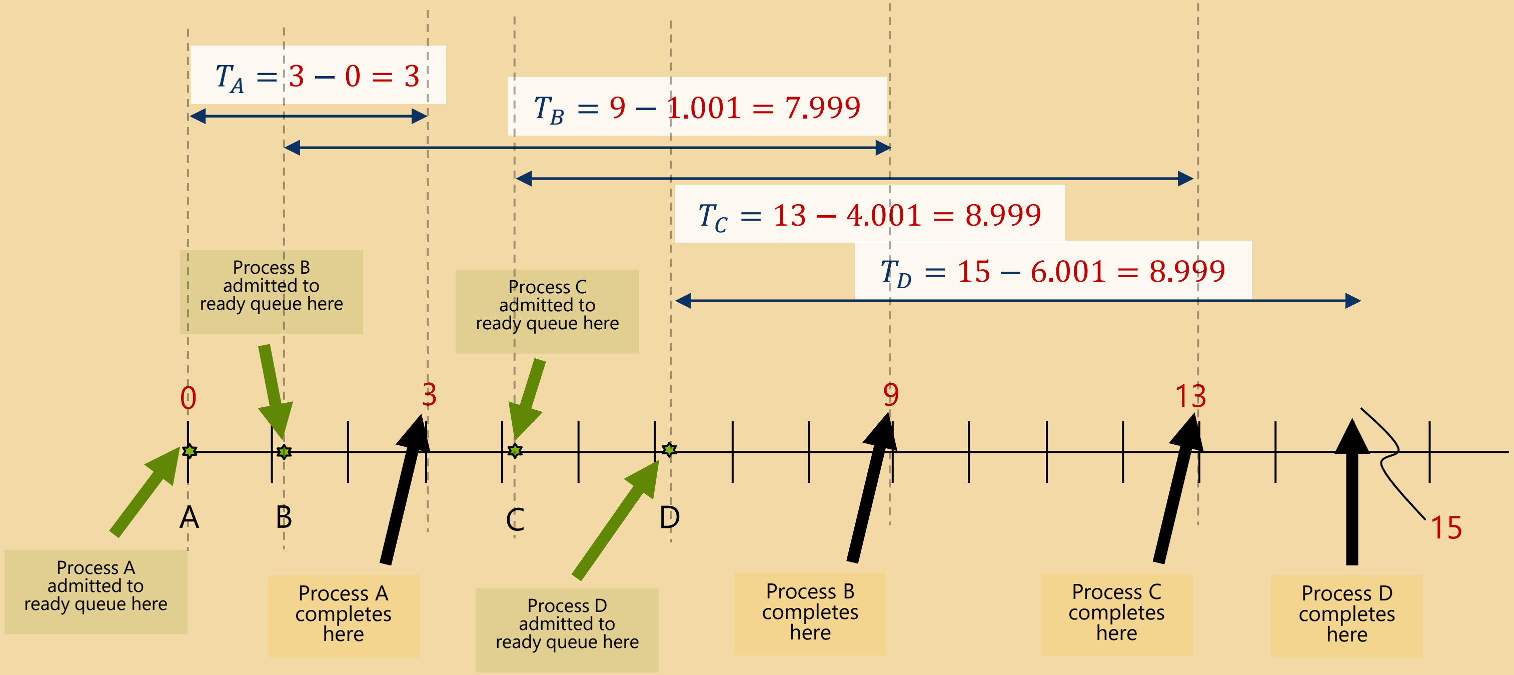


$R_B = 3 - 1.001$
$= 1.999$
$R_C = 9 - 4.001$
$= 4.999$
$R_D = 13 - 6.001$
$= 6.999$
$R_A = 0$
Process A is first assigned to CPU here
Process B is first assigned to CPU here
Process C is first assigned to CPU here
Process D is first assigned to CPU here
0
3
9
13
15
1.001
4.001
6.001
3
A
6
B
4
C
2
D

$$\text{Average response time} = \frac{R_A+R_B+R_C+R_D}{4} = \frac{0+1.999+4.999+6.999}{4} = 3.499 \text{ time units}$$

FCFS : Turnaround Time

Turnaround times for each of the processes (denoted T_i for process i). This is the length of time between when a process first enters the ready queue and when it successfully finishes executing


$T_A = 3 - 0 = 3$

$T_B = 9 - 1.001 = 7.999$

$T_C = 13 - 4.001 = 8.999$

$T_D = 15 - 6.001 = 8.999$

Process B admitted to ready queue here
Process C admitted to ready queue here
0
3
9
13
A
B
C
D
15
Process A admitted to ready queue here
Process A completes here
Process D admitted to ready queue here
Process B completes here
Process C completes here
Process D completes here

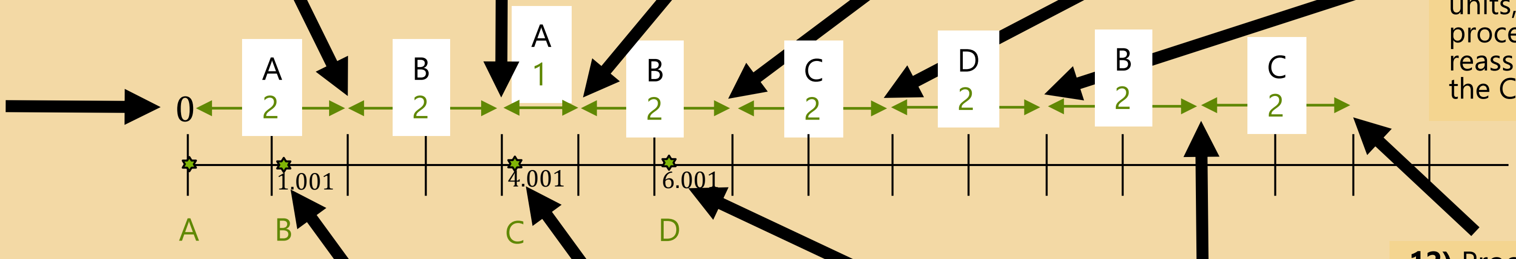
$$\text{Average turnaround time} = \frac{T_A+T_B+T_C+T_D}{4} = \frac{3+7.999+8.999+8.999}{4} = 7.25 \text{ time units}$$

Round Robin

- Decision mode: preemptive, based on hardware clock/timer;
- Selection function: under the Round Robin policy and the assumptions on slide 3, the steps are:
  - When a process is assigned to the CPU, it runs for a fixed amount of time (assume this is 2 units of time) called a *quantum* of time, or a *time slice*;
  - If the process does not terminate before the time quantum elapses, the process rejoins the ready queue. Otherwise, it terminates and yields the CPU;
  - the next process assigned to the CPU is the process that has been in the ready queue the longest. To break a tie, use FCFS;

Round Robin

1) Under Round Robin with a *time slice* of 2 units, process A is assigned to the CPU first and runs for 2 units of time



2) Process B arrives in the ready queue while process A is running

3) Process B is assigned to the CPU after process A has run for 2 time units

4) Process A is reassigned to the CPU after process B has run for 2 time units

5) Process C joins process B in the ready queue, while process A is running

6) Process A completes only after 1 time unit, since it needs 3 time units in total. Process B is reassigned to the CPU

7) Process D joins process C in the ready queue, while process B is running

8) After process B has run for 2 time units, process C is assigned to the CPU

9) After process C has run for 2 time units, process D is assigned to the CPU

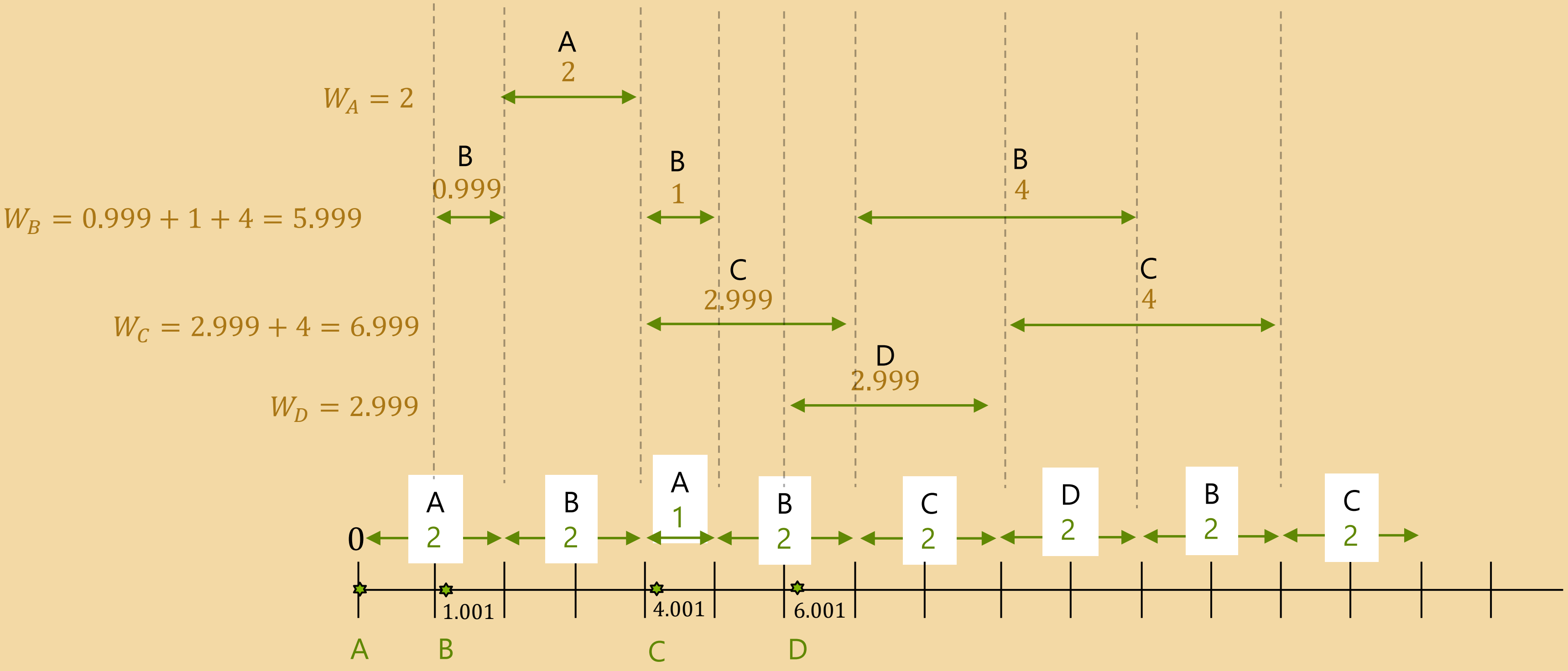
10) Process D completes after 2 time units, and process B is reassigned to the CPU

11) Process B completes after 2 time units, and process C is reassigned to the CPU

12) Process C completes after 2 time units

Round Robin: Wait Time

Wait times for each process (denoted as W_i for process i). This is the total time the process spends waiting to be assigned to the CPU.



$W_A = 2$

$W_B = 0.999 + 1 + 4 = 5.999$

$W_C = 2.999 + 4 = 6.999$

$W_D = 2.999$

Average wait time = $\frac{W_A+W_B+W_C+W_D}{4} = \frac{2+5.998+6.999+2.999}{4} = 4.499$ time units

- Now attempt the two tasks in the “instructions” slide 2. To help get you started, the definitions of SPN and SRT (as they apply to this example) are given on the next two slides.
- If needed, collaborate with each other in groups to figure out the pictures;
- The observant among you will notice that we have not illustrated the average waiting time for the FCFS case. The very observant among you will notice that we have ;-). Can you see why?

Shortest Process Next (SPN)

- Decision mode: non-preemptive – so, with the assumptions in this example, a process will run on the CPU till completion;
- Selection function: under the SPN policy and the assumptions on slide 3, the steps are:
  - When a process is assigned to the CPU, it runs till it terminates;
  - When no process is assigned to the CPU (e.g. when a running process terminates), the process in the ready queue with the shortest CPU burst time is assigned to the CPU. To break a tie, use FCFS;

Shortest Remaining Time (SRT)

- Decision mode: preemptive version of SPN – CPU could be reassigned whenever a new process enters the ready queue;
- Selection function: under the SRT policy and the assumptions on slide 3, the steps are:
  - When a process is assigned to the CPU, it runs until it either terminates or a new process joins the ready queue;
  - When a process terminates or a new process joins the ready queue, the process assigned to the CPU is the process with the shortest remaining CPU burst time. To break a tie, use FCFS;