**PA1 : OS Processes Simulation**

Please write a Java program to simulate the management of OS processes in the system.

You need to create a process-control table (**PCT**) of 7 rows (to manage 7 processes) and 5 columns. The 5 columns are:

1. process id: such as 1234, 4567.

2. status: such as running, waiting, ready, terminated.

3. cpu time used so far: such as 0, 1, 2, 3, …. The initial value is 0.

4. job priority: such as top, high, low, medium. PA1 does not pay attention to the job priority.

5. current instruction counter: such as 0, 1, 2, 3, …, 11. The initial value is -1 (meaning not running yet).

You need to create a process-instruction table (**PIT**) of 7 rows (for those 7 processes) and 12 columns. Each column shows the instruction to be executed (by the OS). The instruction can be compute-1, compute-2, compute-3, input-1, input-2, output-1, output-2, or **finish** (for example). Each process can have up to 12 instructions to be executed with the last instruction being “finish” normally.

You need to create a global (i.e., **static** in Java) variable called **CPUtimer**, which will track the execution of those 7 processes to run through each process’ instructions from PIT column 0 until column 11 or “finish” is encountered (whichever comes first).

You need to preload the data into the **PCT**, which is a String array of 7 by 5. You also need to preload the data into the **PIT**, which is a String array of 7 by 12.

When user issues “run” command, you execute the next instruction of each process. You need to keep track of the current instruction counter for each process. For the first “run”, you will execute the first instruction of each process. And so on until all 7 processes are terminated. When the “**finish**” instruction is executed, you need to change the status of this process to “terminated”, and you should not do anything for this process afterward. After the instruction 11 is executed for a process, that process must be terminated as well.

Your program allows the user to enter some commands as follows:

1. **run** -- “run” command is asking your program to execute all active processes one CPUtime. You must show what you have run for all processes. For the first run, your output will be “Run process 1 instruction 1; Run process 2 instruction 1; …”. After a few runs, your output may be “Run process 5 instruction 4;”. Finally, your output for run command may be “There is no instruction to run for any process!”.

2. **show pct** -- To show the complete contents of the process control table (PCT).

3. **show pit** -- To show the complete contents of the process instruction table (PIT).

4. **ps** -- To show the complete contents of PCT for only *active* processes. A process is *active* if it is not terminated yet.

5. **help** – To list all the available commands with brief explanations.

You must test your program 3 times with different PCT and PIT contents for each test. For each test, you may issue commands in this order for example: show pct; show pit; run; run; ps; run; show pct; show pit; run; run; run; show pct; show pit; ps; …

===========================================================================.

**How to submit your Lab or Project Assignment (PA)?**

(1) Each program must be well-documented with block comments and proper line comments. The

beginning of each program must have a block comment to show your name, date, and purpose.

The following is an example of block and line comments.

/\*

Author: Dr. Simon Lin

Date: 1/10/2016

Purpose: To check whether a string is a palindrome.

\*/

int yearCount = 0 ; // yearCount to count # of years being processed so far

(2) You must submit the following items as attachments through sakai.apu.edu.

(a) All source programs (i.e., all **.java** files), and

(b) One WORD document (i.e., **.doc** or **.docx** file) containing all source programs’ listing and the

output of at least **3** test runs for each program (if input can be changed from one test to another).

You may use snapshots or copy-and-paste method to insert those input/output onto your WORD

document. The WORD document must have header and footer on each page. The **header** must

contain the course name, lab/project id, and your full name. The **footer** must contain the page number

in the format of “Page 1 of 5” (for example) like this Word document.

==========================================================================.

Grading Rubric:

You got \_\_\_ points out of 100 for **CS250 PA1**. Thank you for your excellent/good work.

[ ] -10 points for each day late.

[ ] 30 points – Your program must be fully tested at least 3 times with their complete output shown on your Word document.

[ ] 10 points – Your program must be well-documented.

[ ] 30 points – You must follow the program specification to develop your programs properly and completely.

[ ] 20 points – You must submit your WORD document.

[ ] -5 points if your program did not have block comment to show your name, date, and purpose.

[ ] 5 points – Your WORD document must show “**CS250 PA1**” and your full name on the **header**.

[ ] 5 points – Your WORD document must be page-numbered on the footer with format “Page 2 of 15” (for example).

==================================================================================.

Please use this Word document as a template for your Word document to be submitted. Please delete everything above.

Please copy all your source programs into here:

/\*

Author: JIE GAO

Date: 2/24/2016

Purpose: Using JAVA to simulate OS processes.

\*/

**import** java.util.Scanner;

**public** **class** OSProcessesSimulation {

**public** **static** **void** main(String[]args){

*help*();

System.***out***.println();

Scanner scan=**new** Scanner(System.***in***);

String input="";

**int**[][] PCT={{1001,1,0,1,0},

{1002,1,0,1,0},

{1003,1,0,1,0},

{1004,1,0,1,0},

{1005,1,0,1,0},

{1006,1,0,1,0},

{1007,1,0,1,0}};

String[][] PIT={{"game1", "game2", "game3", "game4", "game5", "game6", "game7", "game8", "game9", "save", "exit", "finsh"},

{"step1", "step2", "step3", "step4", "step5", "step6", "finsh"},{"input1", "input2", "input3", "input4", "finsh"},

{"output1", "output2", "output3", "output4", "finsh"},{"print 100 pages", "finsh"},{"install", "restart"},

{"compute1", "compute2", "compute3", "compute4", "compute5", "compute6", "compute7", "compute8", "compute9", "finsh"}};

**while**(**true**){

System.***out***.println("Enter your command:");

input=scan.nextLine();

**if**(input.equals("run"))

*run*(PCT,PIT);

**else** **if**(input.equals("ps"))

*ps*(PCT,PIT);

**else** **if**(input.equals("show pct"))

*showPCT*(PCT);

**else** **if**(input.equals("show pit"))

*showPIT*(PIT, PCT);

**else** **if**(input.equals("exit"))

System.*exit*(1);

**else** **if**(input.equals("help"))

*help*();

**else**

System.***err***.println("Invalid input.");

System.***out***.println();

}

}

**public** **static** **void** help(){

System.***out***.println("-help-");

System.***out***.println("help – to show commands");

System.***out***.println("show pct – to show the detail of Process Control Table (PCT)");

System.***out***.println("show pit – to show the detail of Process Instruction Table (PIT)");

System.***out***.println("run – to run one CPUtime for each active process");

System.***out***.println("ps – to show the PCT detail of all active processes");

}

**public** **static** **void** run(**int**[][]pct,String[][]pit){

System.***out***.println();

**if**(pct[0][1]==4&&pct[1][1]==4&&pct[2][1]==4&&pct[3][1]==4

&&pct[4][1]==4&&pct[5][1]==4&&pct[6][1]==4)

System.***out***.println("There is no instruction to run for any process!");

**else**{

**for**(**int** i=0;i<pct.length;i++){

**if**(pct[i][1]!=4){

System.***out***.println("Process "+pct[i][0]+":"+pit[i][pct[i][2]]);

pct[i][2]++;

pct[i][4]++;

}

**if**(pct[i][2]>=pit[i].length){

pct[i][1]=4;

}

}

}

}

**public** **static** **void** ps(**int**[][]pct,String[][]pit){

System.***out***.println();

**if**(pct[0][1]==4&&pct[1][1]==4&&pct[2][1]==4&&pct[3][1]==4

&&pct[4][1]==4&&pct[5][1]==4&&pct[6][1]==4)

System.***out***.println("There is no instruction to run for any process!");

**else**{

System.***out***.println("Process-id Status CPU-time-used Jop-priority Current-instruction-counter");

**for**(**int** i=0;i<pct.length;i++){

**if**(pct[i][1]!=4){

System.***out***.println(" "+pct[i][0]+" running "+pct[i][2] + " top " + pct[i][4]);

}

}

}

}

**public** **static** **void** showPCT(**int**[][]pct){

System.***out***.println("Process-id Status CPU-time-used Jop-priority Current-instruction-counter");

**for**(**int** i=0;i<pct.length;i++){

System.***out***.println();

**for**(**int** j=0;j<pct[i].length;j++){

**if**(j==1){

**if**(pct[i][j]==1)

System.***out***.print(" running ");

**else** **if**(pct[i][j]==4)

System.***out***.print(" terminated ");

}

**else** **if**(j==3){

System.***out***.print(" top ");

}

**else**

System.***out***.print(" "+pct[i][j]+" ");

}

}

System.***out***.println();

}

**public** **static** **void** showPIT(String[][]pit, **int**[][]pct){

System.***out***.println();

**for**(**int** i=0;i<pit.length;i++){

System.***out***.println();

**for**(**int** j=0;j<pit[i].length;j++){

System.***out***.print("process"+pct[i][0]+": "+pit[i][j]+", ");

}

}

System.***out***.println();

}

}

Please copy all your testing output into here:

-help-

help – to show commands

show pct – to show the detail of Process Control Table (PCT)

show pit – to show the detail of Process Instruction Table (PIT)

run – to run one CPUtime for each active process

ps – to show the PCT detail of all active processes

Enter your command:

show pct

Process-id Status CPU-time-used Jop-priority Current-instruction-counter

1001 running 0 top 0

1002 running 0 top 0

1003 running 0 top 0

1004 running 0 top 0

1005 running 0 top 0

1006 running 0 top 0

1007 running 0 top 0

Enter your command:

show pit

process1001: game1, process1001: game2, process1001: game3, process1001: game4, process1001: game5, process1001: game6, process1001: game7, process1001: game8, process1001: game9, process1001: save, process1001: exit, process1001: finsh,

process1002: step1, process1002: step2, process1002: step3, process1002: step4, process1002: step5, process1002: step6, process1002: finsh,

process1003: input1, process1003: input2, process1003: input3, process1003: input4, process1003: finsh,

process1004: output1, process1004: output2, process1004: output3, process1004: output4, process1004: finsh,

process1005: print 100 pages, process1005: finsh,

process1006: install, process1006: restart,

process1007: compute1, process1007: compute2, process1007: compute3, process1007: compute4, process1007: compute5, process1007: compute6, process1007: compute7, process1007: compute8, process1007: compute9, process1007: finsh,

Enter your command:

run

Process 1001:game1

Process 1002:step1

Process 1003:input1

Process 1004:output1

Process 1005:print 100 pages

Process 1006:install

Process 1007:compute1

Enter your command:

run

Process 1001:game2

Process 1002:step2

Process 1003:input2

Process 1004:output2

Process 1005:finsh

Process 1006:restart

Process 1007:compute2

Enter your command:

ps

Process-id Status CPU-time-used Jop-priority Current-instruction-counter

1001 running 2 top 2

1002 running 2 top 2

1003 running 2 top 2

1004 running 2 top 2

1007 running 2 top 2

Enter your command:

run

Process 1001:game3

Process 1002:step3

Process 1003:input3

Process 1004:output3

Process 1007:compute3

Enter your command:

show pct

Process-id Status CPU-time-used Jop-priority Current-instruction-counter

1001 running 3 top 3

1002 running 3 top 3

1003 running 3 top 3

1004 running 3 top 3

1005 terminated 2 top 2

1006 terminated 2 top 2

1007 running 3 top 3

Enter your command:

show pit

process1001: game1, process1001: game2, process1001: game3, process1001: game4, process1001: game5, process1001: game6, process1001: game7, process1001: game8, process1001: game9, process1001: save, process1001: exit, process1001: finsh,

process1002: step1, process1002: step2, process1002: step3, process1002: step4, process1002: step5, process1002: step6, process1002: finsh,

process1003: input1, process1003: input2, process1003: input3, process1003: input4, process1003: finsh,

process1004: output1, process1004: output2, process1004: output3, process1004: output4, process1004: finsh,

process1005: print 100 pages, process1005: finsh,

process1006: install, process1006: restart,

process1007: compute1, process1007: compute2, process1007: compute3, process1007: compute4, process1007: compute5, process1007: compute6, process1007: compute7, process1007: compute8, process1007: compute9, process1007: finsh,

Enter your command:

run

Process 1001:game4

Process 1002:step4

Process 1003:input4

Process 1004:output4

Process 1007:compute4

Enter your command:

run

Process 1001:game5

Process 1002:step5

Process 1003:finsh

Process 1004:finsh

Process 1007:compute5

Enter your command:

run

Process 1001:game6

Process 1002:step6

Process 1007:compute6

Enter your command:

show pct

Process-id Status CPU-time-used Jop-priority Current-instruction-counter

1001 running 6 top 6

1002 running 6 top 6

1003 terminated 5 top 5

1004 terminated 5 top 5

1005 terminated 2 top 2

1006 terminated 2 top 2

1007 running 6 top 6

Enter your command:

show pit

process1001: game1, process1001: game2, process1001: game3, process1001: game4, process1001: game5, process1001: game6, process1001: game7, process1001: game8, process1001: game9, process1001: save, process1001: exit, process1001: finsh,

process1002: step1, process1002: step2, process1002: step3, process1002: step4, process1002: step5, process1002: step6, process1002: finsh,

process1003: input1, process1003: input2, process1003: input3, process1003: input4, process1003: finsh,

process1004: output1, process1004: output2, process1004: output3, process1004: output4, process1004: finsh,

process1005: print 100 pages, process1005: finsh,

process1006: install, process1006: restart,

process1007: compute1, process1007: compute2, process1007: compute3, process1007: compute4, process1007: compute5, process1007: compute6, process1007: compute7, process1007: compute8, process1007: compute9, process1007: finsh,

Enter your command:

ps

Process-id Status CPU-time-used Jop-priority Current-instruction-counter

1001 running 6 top 6

1002 running 6 top 6

1007 running 6 top 6

Enter your command:

run

Process 1001:game7

Process 1002:finsh

Process 1007:compute7

Enter your command:

run

Process 1001:game8

Process 1007:compute8

Enter your command:

show pct

Process-id Status CPU-time-used Jop-priority Current-instruction-counter

1001 running 8 top 8

1002 terminated 7 top 7

1003 terminated 5 top 5

1004 terminated 5 top 5

1005 terminated 2 top 2

1006 terminated 2 top 2

1007 running 8 top 8

Enter your command:

run

Process 1001:game9

Process 1007:compute9

Enter your command:

run

Process 1001:save

Process 1007:finsh

Enter your command:

show pct

Process-id Status CPU-time-used Jop-priority Current-instruction-counter

1001 running 10 top 10

1002 terminated 7 top 7

1003 terminated 5 top 5

1004 terminated 5 top 5

1005 terminated 2 top 2

1006 terminated 2 top 2

1007 terminated 10 top 10

Enter your command:

show pit

process1001: game1, process1001: game2, process1001: game3, process1001: game4, process1001: game5, process1001: game6, process1001: game7, process1001: game8, process1001: game9, process1001: save, process1001: exit, process1001: finsh,

process1002: step1, process1002: step2, process1002: step3, process1002: step4, process1002: step5, process1002: step6, process1002: finsh,

process1003: input1, process1003: input2, process1003: input3, process1003: input4, process1003: finsh,

process1004: output1, process1004: output2, process1004: output3, process1004: output4, process1004: finsh,

process1005: print 100 pages, process1005: finsh,

process1006: install, process1006: restart,

process1007: compute1, process1007: compute2, process1007: compute3, process1007: compute4, process1007: compute5, process1007: compute6, process1007: compute7, process1007: compute8, process1007: compute9, process1007: finsh,

Enter your command:

ps

Process-id Status CPU-time-used Jop-priority Current-instruction-counter

1001 running 10 top 10

Enter your command:

run

Process 1001:exit

Enter your command:

run

Process 1001:finsh

Enter your command:

run

There is no instruction to run for any process!

Enter your command:

ps

There is no instruction to run for any process!

Enter your command:

show pct

Process-id Status CPU-time-used Jop-priority Current-instruction-counter

1001 terminated 12 top 12

1002 terminated 7 top 7

1003 terminated 5 top 5

1004 terminated 5 top 5

1005 terminated 2 top 2

1006 terminated 2 top 2

1007 terminated 10 top 10

Enter your command:

help

-help-

help – to show commands

show pct – to show the detail of Process Control Table (PCT)

show pit – to show the detail of Process Instruction Table (PIT)

run – to run one CPUtime for each active process

ps – to show the PCT detail of all active processes

Enter your command:

exit