Operating Systems (15841, Spring 2016)
Project

Professor Joongheon Kim

School of Computer Science and Engineering, Chung-Ang University, Seoul, Republic of Korea

Overview

- Theory Projects
  - Improving the Banker's Algorithm
  - Improving the Paging Algorithm
  - Paper Reading/Presentation/Discussion

- Implementation Projects
  - Process Execution Simulator
  - Banker's Algorithm Simulator
  - Paging Simulator (with Physical/Logical Address Mapping)

Theory Project

- **[Project-T1]** Improve the Banker's Algorithm in your own way
  - You have to define the drawbacks of the currently existing Banker's algorithm and propose the solution approaches to the drawbacks.
  - You have to submit intermediate report; and final report/poster.

- **[Project-T2]** Improve the Paging Algorithm in your own way
  - You have to define the drawbacks of the currently existing paging algorithm and propose the solution approaches to the drawbacks.
  - You have to submit intermediate report; and final report/poster.

- **[Project-T3]** Paper Reading/Presentation/Discussion
  - You have to read given papers.
  - You have to submit final summary report and poster.

Schedule

- For **[Project-T1]** and **[Project-T2]** (new idea proposals)
  - [May 20th (Friday)] Intermediate Progress **Report** Submission (for each topic)
    - The report should contain following contents:
      - Introduction and Motivation
      - The Proposed Algorithm
      - Performance Evaluation Approaches
    - Each team needs to meet Prof. Kim in order to present the algorithm (by appointment)
  - [June 6th (Monday)] Final **Report** and **Poster** Submission (for each topic)
    - The report should contain following contents:
      - Introduction and Motivation
      - The Proposed Algorithm
      - Performance Evaluation and Future Research Directions
    - Each team needs to meet Prof. Kim in order to present the algorithm (by appointment)
- For **[Project-T3]** (for paper reading/presentation/discussion)
  - [June 6th (Monday)] Final **Report** and **Poster** Submission (for each topic)
    - The report should contain following contents:
      - Introduction and Motivation
      - The Proposed Algorithm
      - Performance Evaluation and Discussion (In "discussion", you have to clearly mention the disadvantages of the paper.)

Theory Project - List of Papers **(Systems and Platforms)**

- S. Yoo, Y. Shim, S. Lee, S.-A. Lee, and J. Kim, "**A Case for Bad big.LITTLE Switching: How to Scale Power-Performance in SI-HMP**," in Proceedings of the ACM Symposium on Operating Systems Principles (SOSP) Workshop on Power-Aware Computing and Systems (HotPower), Monterey, CA, USA, 4 October 2015.
- J. Paek, J. Kim, and R. Govindan, "**Energy-Efficient Rate-Adaptive GPS-based Positioning for Smartphones**," in Proceedings of the ACM International Conference on Mobile Systems, Applications, and Services (MobiSys), San Francisco, CA, USA, 15 - 18 June 2010.
- C. Noyes, "**BitAV: Fast Anti-Malware by Distributed Blockchain Consensus and Feedforward Scanning**," http://arxiv.org/pdf/1601.01405.pdf

Theory Project - List of Papers **(Networks)**

- M.-R. Ra, J. Paek, A.B. Sharma, R. Govindan, M.H. Krieger, and M.J. Neely, "**Energy-Delay Tradeoffs in Smartphone Applications**," in Proceedings of the ACM International Conference on Mobile Systems, Applications, and Services (MobiSys), San Francisco, CA, USA, 15 - 18 June 2010.
- J. Kim, G. Caire, and A.F. Molisch, "**Quality-Aware Streaming and Scheduling for Device-to-Device Video Delivery**," IEEE/ACM Transactions on Networking, (Published Online).

Implementation Project

- **[Project-I1]** Process Execution Simulator
- **[Project-I2]** Banker's Algorithm Simulator
- **[Project-I3]** Paging Simulator (with Physical/Logical Address Mapping)

- For all the four projects,
  - Your software should read a configuration file.
  - Your software should conduct desired functionalities.
  - Your software should generate a final result file.

- Submission Schedule
  - Submission Due: June 6th, 11:59pm
  - Submission Files
    - Input file
    - Source code
    - Output file

[Project-I1] Process Execution Simulator

- You have to submit three files
  - Names: configuration_process.xml, simulation_process.c (or cpp/java), finalresult_process.xml
    - **configuration_process.xml**
      - This file contains all required information for process execution including following items
        - The number of Program Counters
        - The number of Processes
        - The starting and ending addresses of each process
        - The starting and ending addresses of each process
        - And whatever you want
    - **simulation_process.c**
      - You can use C, C++, Java, Python, Lisp, or Scheme
    - **finalresult_process.xml**
      - This contains traces for each process execution.
      - There is no specific format for this. But, you have to explain how this file can be understood in final report.
    - Note
      - The configuration and final result files can be xml. But, you can define your own.
      - You should submit one report which shows your software architecture, compilation procedure, final result file understanding.
      - For evaluation, I will change the values in **configuration_process.xml**.

[Project-I2] Banker's Algorithm Simulator

- You have to submit three files
  - Names: configuration_banker.xml, simulation_banker.c (or cpp/java), finalresult_banker.xml
    - **configuration_banker.xml**
      - This file contains all required information for the banker's algorithm including following items
        - Allocation Matrix, Maximum Matrix (assume that we can have infinite types of resources and infinite number of processes)
        - Competition Priority among processes
        - And whatever you want
    - **simulation_banker.c**
      - You can use C, C++, Java, Python, Lisp, or Scheme
    - **finalresult_banker.txt**
      - This contains traces for process scheduling.
      - There is no specific format for this. But, you have to explain how this file can be understood in final report.
      - Sample example for the trace file format is in the next page (obtained from our HW1 solution)
    - Note
      - The configuration and final result files can be xml. But, you can define your own.
      - You should submit one report which shows your software architecture, compilation procedure, final result file understanding.
      - For evaluation, I will change the values in **configuration_banker.xml**.

[Project-I2] Banker's Algorithm Simulator

- Sample for Final Result File

[Step 1]
- Total [11,7,7]
- Available: [7,5,2]
- P0 → A: [0,1,0], N: [7,4,3]
- P1 → A: [1,0,0], N: [2,2,2]
- P2 → A: [2,0,2], N: [7,0,0]
- P3 → A: [1,1,1], N: [1,1,1]
- P4 → A: [0,0,2], N: [5,3,1]

P1,P2,P3,P4 can be processed.
P1 is selected to be processed.
(i) Allocate [2,2,2] to P1
(ii) P1 completes its work and returns resources

[Step 2]
- Available: [8,5,2]
- P0 → A: [0,1,0], N: [7,4,3]
- P2 → A: [2,0,2], N: [7,0,0]
- P3 → A: [1,1,1], N: [1,1,1]
- P4 → A: [0,0,2], N: [5,3,1]

P2,P3,P4 can be processed.
P2 is selected to be processed.
(i) Allocate [7,0,0] to P2
(ii) P2 completes its work and returns resources

[Step 3]
- Available: [10,5,4]
- P0 → A: [0,1,0], N: [7,4,3]
- P3 → A: [1,1,1], N: [1,1,1]
- P4 → A: [0,0,2], N: [5,3,1]

P0,P3,P4 can be processed.
P2 is selected to be processed.
(i) Allocate [7,4,3] to P0
(ii) P0 completes its work and returns resources

[Step 4]
- Available: [10,6,4]
- P3 → A: [1,1,1], N: [1,1,1]
- P4 → A: [0,0,2], N: [5,3,1]

P3,P4 can be processed
P3 is selected to be processed.
(i) Allocate [1,1,1] to P1
(ii) P1 completes its work and returns resources

[Step 5]
- Available: [11,7,5]
- P4 → A: [0,0,2], N: [5,3,1]

P4 is selected to be processed.
(i) Allocate [5,3,1] to P1
(ii) P1 completes its work and returns resources

[Final]
Processing order: P1, P2, P0, P3, P4

[Project-I3] Paging Simulator (with Physical/Logical Address Mapping)

- You have to submit three files
  - Names: configuration_paging.xml, simulation_paging.c (or cpp/java), finalresult_paging.xml
    - **configuration_paging.xml**
      - This file contains all required information for paging operation including following items
        - Process related information (numbers, sizes, and segment sizes, process scheduling priorities)
        - Page size, offset size, address size (these sizes will be equivalent to the textbook configuration)
        - Sequence of process allocation and corresponding information (your paging software should work with at least 20 processes and shows the step-by-step procedure for the paging operation in **finalresult_paging.txt** file).
        - And whatever you want.
    - **simulation_paging.c**
      - You can use C, C++, Java, Python, Lisp, or Scheme
      - You have to implement physical/logical address mapping functions in your code. Both addresses should be presented in **finalresult_paging.txt** trace file.
    - **finalresult_paging.txt**
      - This contains traces for paging operation.
      - There is no specific format for this. But, you have to explain how this file can be understood in final report.
    - Note
      - The configuration and final result files can be xml. But, you can define your own.
      - You should submit one report which shows your software architecture, compilation procedure, final result file understanding.
      - For evaluation, I will change the values in **configuration_paging.xml**.