Memory-Manager

Memory-Manager

Introduction

PPT

Inplemented memory manager by C language, and
- Translation Lookaside Buffer - Random - LRU (Least Recently Used) - Page Replacement Policy - FIFO (First In, First Out) - Clock (Second Chance replacement) - Frame Allocation Policy - Global - Local

three kind policies would be changed, in order to analyze their pros and cons.

Data Analysis

Experiment Data

example 1

policy

```
TLB Replacement Policy: RANDOM
Page Replacement Policy: CLOCK
Frame Allocation Policy: LOCAL
Number of Processes: 2
Number of Virtual Page: 128
Number of Physical Frame: 64
```

result

```
Process A, Effect Access Time = 169.817
Process A, Page Fault Rate = 0.774
Process B, Effect Access Time = 169.704
Process B, Page Fault Rate = 0.694
```

example 2

policy

```
TLB Replacement Policy: LRU
Page Replacement Policy: CLOCK
Frame Allocation Policy: LOCAL
Number of Processes: 2
Number of Virtual Page: 128
Number of Physical Frame: 64
```

result

```
Process A, Effect Access Time = 164.980
Process A, Page Fault Rate = 0.774
Process B, Effect Access Time = 163.522
Process B, Page Fault Rate = 0.694
```

example 3

policy

```
TLB Replacement Policy: LRU
Page Replacement Policy: FIFO
Frame Allocation Policy: LOCAL
Number of Processes: 2
Number of Virtual Page: 128
Number of Physical Frame: 64
```

result

```
Process A, Effect Access Time = 164.980
Process A, Page Fault Rate = 0.774
Process B, Effect Access Time = 163.144
Process B, Page Fault Rate = 0.700
```

example 4

policy

```
TLB Replacement Policy: LRU
Page Replacement Policy: CLOCK
Frame Allocation Policy: GLOBAL
Number of Processes: 2
Number of Virtual Page: 128
Number of Physical Frame: 64
```

result

```
Process A, Effect Access Time = 164.758
Process A, Page Fault Rate = 0.723
Process B, Effect Access Time = 163.709
Process B, Page Fault Rate = 0.665
```

Analysis

1. TLB policy

> 當 Translation Lookaside Buffer 滿時，決定應該要從何處開始將上方的 element (包含此 process virtual page 的 physical frame) 移出，並代換為新的 element

下方 example 1 & example 2 可以看出，如果使用 Random ，那 effect access time 會降低，因此 **LRU (Least Recently Used) 相對於 Random 在 Effect Access Time 的效率考慮上會是較佳的選擇**

2. Page Replacement Policy

> 當 Physical Memory 滿時，決定應該從何處開始將 element (包含此 physical frame 的 process 與 virtual page) 移出，並代換為新的 element

從下方 example 1 & example 3 可以看出，**FIFO & Clock 相距不大，所以可以解讀為兩者的效率相差不遠**，無法在此範例測資中看出大的不同，或許可以利用大量數據測資，再進一步觀察

3. Frame Allocation Policy

> 當 Physical Memory 滿時，決定能否將不屬於 current process 的 physical frame 替換

會被每個 OS 的系統所決定，並不像是手動調整的人為因素，也許系統的安全有關。但如果單純觀察結果，可以看出，**Effect Access Time 和 Page fault rate 會大幅度上升**，因為 Local access Physical memory 的權限增加，不同的 process 即不能 access 到不同的 process frame