操作系统算法竞赛
A ten-day Hackathon

要求与提交

题目

- demand paging 机制下，页面替换算法设计
  - 实现现有典型替换算法 .
  - 设计一种新型的替换算法
  - 开展实验验证算法性能

- 10 天完成
  - 截止时间： 12.13
  - 汇报 PPT （汇报时间 12.15 ）
  - 算法设计技术报告（ doc ， pdf)

Page Replacement

- Demand paging
  - Start process with no pages loaded
  - When a process faults & memory is full
  - Find a victim frame in memory
  - Swap it out

Some Basic Algorithms

- Random: pick any page at random (works surprisingly well!).
- FIFO: throw out the page that has been in memory the longest. The idea is to be fair, give all pages equal residency.
- MIN （OPT） : naturally, the best algorithm arises if we can predict the future.
- LFU: use the frequency of past references to predict the future.
- LRU: use the order of past references to predict the future.

进一步的思考

- 替换算法一般考虑两种因素
  - Recency： 访问时间
  - Frequency： 访问频率
- 在此基础上可以进一步组合，扩展
  - 例如区分不同页面，设置不同替换策略
- 可以参考 Cache Replacement Championship
  - https://www.sigarch.org/call-contributions/the-2nd-cache-replacement-championship/

实验开展

- 基本的算法比较至少实现下面 4 种与自己的算法
  - FIFO， MIN， LRU， Second chance
  - 自己的算法
- frame number 变化
  - 基本： 3， 4， 5， 6， 10， 20， 30， 40， 50, 100
  - 可以更多数字
- 自己设计的算法本身参数变化对算法性能影响
  - 比如统计过去一段时间的频率，时间窗口大小
- 不同的 trace 下，算法的性能

页面访问的 trace

- 4 个基本数据集（必须做的）
  - trace 内容为每行一个页面号（不是内存地址）
  - 直接根据页面号比较即可

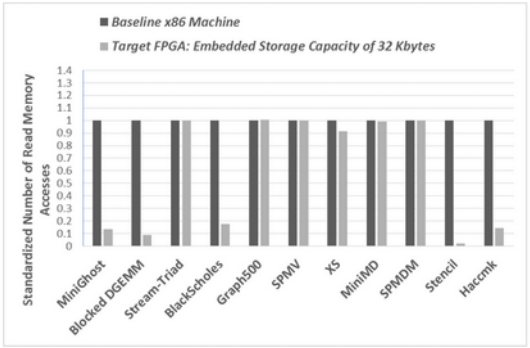
  - qsort: quicksort program
    - 524 references, and 192 unique pages
  - lu: LU decomposition algorithm
    - 41832 references, and 531 unique pages
  - mm32: matrix multiply 32x32
    - 67235 references to 1667 unique pages
  - mm16: matrix multiply 16x16
    - 9075 references to 515 unique pages

- 在此基础上可以利用其他 trace （选做）
  - 或许需要额外的格式化处理
  - 例如
    - https://traces.cs.umass.edu/index.php/CpuMem/CpuMem
- 需要能验证出自己设计算法的好处。

实验结果展示

- 实验结果可用表格或者图
  - 同一组数据一种即可
- 表格（比较精确）
  - 具体数字
- 图（可以看到趋势）
  - 把实验的数据用画图工具画出来
  - 例如 matlab， gnuplot， excel
- 每张图和表都需要加文字对其结果趋势进行分析总结。

Benchmark	Baseline Number of x86 Read Accesses	Target Number of FPGA Read Accesses
MiniGhost	4849969	663552
Blocked DGEMM	33554439	1114112
Stream-Triad	2001	2000
BlackScholes	243728	43008
Graph500	2195185	2202251



选做

- Pre-paging 机制
  - OS guesses in advance which page the process will require and pre-loads them into the memory.
  - 设计一个 prefetching 算法
    - 预测未来访问的页面
    - 提前放进内存，避免缺页
  - 比较带 prefetching 和不带 prefetchig 的缺页次数与硬盘总读取次数

组队

- 组队（需要在周二前完成）
  - 5 人 / 组，如人不够可以少，但不少于 3 人 / 组
  - 填写腾讯文档
    - https://docs.qq.com/sheet/DQlZMZW1DZU10bXd1?tab=BB08J2
- 组队完成后
  - 小组取名
  - 分工查阅资料，趋势分析
  - 设计算法
  - 实现代码
  - 实验效果比较
  - 撰写技术报告和 PPT

汇报 PPT 要求

- 每个小组汇报 15 分钟，必须包含以下内容
  - 封面：题目 + 小组名，题目可体现算法特点
  - 成员：所有成员在一起的合影（不限形式，可以在任何地点，可以讨论时，聚会时等）
  - 成员名字列表，大家的分工贡献
  - 算法设计：
    - 现有算法哪些方面不足
    - 利用了什么特征，设计的思路与实现
    - 带来的优势和可能的开销分析
  - 算法评价效果
    - 不同的典型算法性能比较
    - 不同的访问序列
    - 不同的算法参数（例如 frame 数量等）

技术报告要求

- 每个组都要提交
  - 同小组多人，只需要组长提交一份。
- 格式要求
  - 题目，小组成员与分工
  - 简介
  - 设计的思路
  - 实现细节
  - 实验评估
  - 参考文献
- 提交网站：学习通作业区
  - 文件名： 组名 - 题目 .docx　或者 pdf

评价形式

- 评价依据
  - 算法的性能（基准测试数据的缺页次数）
  - 汇报的效果（设计清晰，流利）

- 老师打分（30%） + 小组互评（70%）
- 小组互评
  - 所有人对每个小组打分（每人一票）

- 得分最高者为最后冠军队
  - 单项奖励：最佳性能，最佳汇报