|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **学校** | 珠海科技学院 |  | **班级** | 22级4班 |  | **科目** | 操作系统 |
| **学院** | 计算机学院 |  | **姓名** | 陈俊伟 |  | **日期** | 2024.6.13 |
| **专业** | 计算机科学与技术 |  | **学号** | 04220411 |  | **实验名称** | 页面置换LRU |

|  |
| --- |
| 实验内容  第1题 编写一个程序，用rand()生成一个随机的页访问序列，该序列长度为320；采用LRU的页面置换策略，计算当页数为32，帧数为4～32时的页面访问命中率。（命中率＝1-页面失效次数／页面访问序列长度。）  命令  #include <stdio.h> #include<unistd.h> #include<time.h> #include<stdlib.h> struct pl\_type {     int pn,pfn,time; };  struct pfc\_struct {     int pn,pfn;     struct pfc\_struct \* next; };  int s[320];                 struct pl\_type pl[32];  struct pfc\_struct pfc[32];  struct pfc\_struct \* freepf\_head;  void initial(int pf); void lru(int pf);  int main() {     int i,total\_pf;      srand(getpid());                for(i=0;i<320;i++)     {         s[i]=rand()%32;     }      for(total\_pf=4;total\_pf<=32;total\_pf++)         {         initial(total\_pf);         lru(total\_pf);     }     return 0; }  void initial(int pf) {     int i;      for(i=0;i<32;i++)               {         pl[i].pn=i;         pl[i].pfn=-1;         pl[i].time=0;     }      for(i=0;i<pf-1;i++)             {         pfc[i].pfn=i;         pfc[i].next=&pfc[i+1];     }     pfc[pf-1].pfn=pf-1;     pfc[pf-1].next=NULL;      freepf\_head=&pfc[0]; }  void lru(int pf) {     int i,j,diseffect=0;            /\*diseffect用来记录页面失效次数\*/     int mintime,minj,present\_time;  /\*present\_time用于计时\*/      for(i=0;i<320;i++)              /\*处理页面访问序列，计算失效数\*/     {         if(pl[s[i]].pfn==-1)        /\*被访问页面不在内存中，失效数加1并将其调入内存\*/         {             diseffect+=1;              if(freepf\_head==NULL)       /\*无空闲帧，则找出距现在最久的被访问页，并将淘汰，将其占用的帧加入空闲帧队列\*/             {                 mintime=32767;                 for(j=0;j<32;j++)                 {                     if(mintime>pl[j].time&&pl[j].pfn!=-1)                     {                         mintime=pl[j].time;                         minj=j;                     }                 }                 freepf\_head=&pfc[pl[minj].pfn];                         pl[minj].pfn=-1;                 pl[minj].time=-1;                 freepf\_head->next=NULL;                         }             pl[s[i]].pfn=freepf\_head->pfn;          /\*为待调入页面分配一帧，并记录下访问时间\*/             pl[s[i]].time=present\_time;             freepf\_head=freepf\_head->next;         }                     else             pl[s[i]].time=present\_time;         /\*若被访页面在内存中则更新访问时间\*/         present\_time++;                         /\*每处理页面访问序列中的一项，计时器加1\*/     }     printf("%d frames %f\n",pf,1-(float)diseffect/320); }  运行结果截图    图1-1 |
| 通过本次实验，深刻学习了页面置换算法LRU |