#include <stdio.h>

#include <stdlib.h>

typedef struct node {

int data;

int frequency;

struct node \*next;

} Node;

void lfu(int page);

void print();

Node \*start = NULL, \*last = NULL;

int pgfault = 0, m, n;

int a[80];

int main() {

int i;

printf("How many frames=");

scanf("%d", &m);

printf("Enter the number of pages:\t");

scanf("%d", &n);

for (i = 0; i < n; i++) {

printf("Enter the page no=%d\t", i);

scanf("%d", &a[i]);

}

for (i = 0; i < n; i++) {

lfu(a[i]);

printf("\n");

print();

}

printf("\nNumber of page faults: %d\n", pgfault);

return 0;

}

void lfu(int page) {

Node \*ptr, \*temp, \*lfuNode, \*prev, \*prevPtr;

int found = 0;

static int cnt = 0;

for (ptr = start; ptr != NULL; ptr = ptr->next) {

if (ptr->data == page) {

ptr->frequency++;

found = 1;

}

}

if (found == 0) {

pgfault++;

if (cnt < m) {

ptr = (Node \*)malloc(sizeof(Node));

ptr->data = page;

ptr->next = NULL;

ptr->frequency = 1;

if (start == NULL)

start = last = ptr;

else {

last->next = ptr;

last = ptr;

}

cnt++;

} else {

lfuNode = start;

temp = start;

prev = start;

while (temp != NULL) {

if (lfuNode->frequency > temp->frequency) {

lfuNode = temp;

prevPtr = prev;

}

prev = temp;

temp = temp->next;

}

lfuNode->data = page;

lfuNode->frequency = 1;

if (lfuNode == last)

return;

if (lfuNode == start) {

start = start->next;

lfuNode->next = NULL;

last->next = lfuNode;

last = lfuNode;

} else {

prevPtr->next = lfuNode->next;

lfuNode->next = NULL;

last->next = lfuNode;

last = lfuNode;

}

}

}

}

void print() {

Node \*ptr;

for (ptr = start; ptr != NULL; ptr = ptr->next) {

printf("%d\t%d\t", ptr->data, ptr->frequency);

}

printf("\n");

}

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<unistd.h>

#include<sys/types.h>

#include<sys/stat.h>

#include<fcntl.h>

#include<dirent.h>

void typeline(char \*s,char \* fn){

int i=0,j=0,cnt=0,n,num;

char c,buff[80];

FILE \*fd;

fd= fopen(fn,"r");

if(fd == NULL)

{

printf("file cannot be opened\n");

return;

}

n=atoi(s);

if (n == 0)

c='a';

else if(n > 0)

c='+';

else

c='-';

switch(c){

case 'a' :while(fgets(buff,80,fd)){

printf("%s",buff);

}

break;

case '+' :cnt = 0;

while(fgets(buff,80,fd)){

printf("%s",buff);

cnt++;

if(cnt==n)

break;

}

break;

case '-' :cnt = 0;

while(fgets(buff,80,fd)){

cnt++;

}

num =cnt + n;

j=0;

fseek(fd,0,SEEK\_SET);

while(fgets(buff,80,fd) && j<num){

j++;

}

while(fgets(buff,80,fd)){

printf("%s",buff);

}

break;

}

}

int main(){

char buff[80],t1[20],t2[20],t3[20],t4[20];

int n,pid,k=0,status;

system("clear");

while(1){

printf("my\_shell$");

fflush(stdin);

fgets(buff,80,stdin);

t1[0]=t2[0]=t3[0]=t4[0]='\0';

n=sscanf(buff,"%s%s%s%s",t1,t2,t3,t4);

if(strcmp(t1,"exit\0")==0){

exit(0);

}

if(strcmp(t1,"typeline\0")==0){

typeline(t2,t3);

}

else if ((pid=fork())==0){

switch(n){

case 1: execlp(t1,t2,NULL);

break;

case 2: execlp(t1,t2,t3,NULL);

break;

case 3: execlp(t1,t2,t3,t4);

break;

}

}

}

}