/\*------------ Linked File Allacation --------------\*/

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

struct file

{

char filename[20];

int filesize;

int startblock;

int endblock;

};

typedef struct file f;

struct node

{

int blockno;

struct node \*next;

};

typedef struct node block;

block \*freelist=NULL,\*allocate=NULL,\*lasta=NULL;

f f1[20];

int no=1,d,size,count=1,countfblock=0;

block \*getblock()

{

block \*temp;

temp=(block \*)malloc(sizeof(block));

temp->blockno=no;

no++;

temp->next=NULL;

return(temp);

}

block \* addblock(block \*ptr)

{

block \*temp;

temp=(block \*)malloc(sizeof(block));

temp->blockno=ptr->blockno;

temp->next=NULL;

return(temp);

}

block \* create()

{

block \*temp,\*last=NULL,\*list=NULL;

int i;

for(i=0;i<d;i++)

{

temp=getblock();

if(list==NULL)

{

list=temp;

last=temp;

}

else

{

last->next=temp;

last=temp;

}

}

return(list);

}

block \*createalloclist()

{

block \*temp,\*ptr=freelist,\*prev;

int i;

f1[count].startblock=ptr->blockno;

for(i=0;i<f1[count].filesize && ptr!=NULL;i++)

{

temp=addblock(ptr);

f1[count].endblock=temp->blockno;

prev=ptr;

freelist=ptr->next;

ptr=ptr->next;

free(prev);

if(allocate==NULL)

{

allocate=temp;

lasta=temp;

}

else

{

lasta->next=temp;

lasta=temp;

}

}

return(allocate);

}

void displaylist(block \*list1)

{

block \*ptr;

for(ptr=list1;ptr!=NULL;ptr=ptr->next)

printf("%d->",ptr->blockno);

}

void countfree(block \*list1)

{

block \*ptr;

for(ptr=list1;ptr->next!=NULL;ptr=ptr->next)

countfblock++;

}

void acceptfile()

{

printf("\nEnter the file name:");

scanf("%s",&f1[count].filename);

printf("\nEnter file size in blocks:");

scanf("%d",&f1[count].filesize);

}

void displayfile()

{

int i;

printf("\nFile name\t\tFile size\t\tstart block\t\tEnd block");

for(i=1;i<=count;i++)

{

printf("\n%s",f1[i].filename);

printf("\t\t\t%d",f1[i].filesize);

printf("\t\t\t%d",f1[i].startblock);

printf("\t\t\t%d",f1[i].endblock);

}

}

int main()

{

int ch,result;

char fname[20];

printf("\nEnter the size of disk in blocks");

scanf("%d",&d);

freelist=create();

while(1)

{

printf("\n1: Allocate space for newly created file.");

printf("\n2: Show used and free space on disk.");

printf("\n3: Exit");

printf("\nEnter the choice");

scanf("%d",&ch);

switch(ch)

{

case 1:

acceptfile();

countfree(freelist);

if(countfblock>=f1[count].filesize)

{

allocate=createalloclist();

displayfile();

count++;

}

else

printf("\nNo sufficient space to allocate");

break;

case 2:

printf("\nFree list:");

displaylist(freelist);

printf("\nAllocated list: ");

displaylist(allocate);

break;

case 3:

exit(0);

}

}

}