(36)

#include <stdio.h>

#include <stdlib.h>

#define MAX\_BLOCKS 10

typedef struct Block {

int next\_block;

} Block;

typedef struct File {

int first\_block;

int last\_block;

} File;

Block disk[MAX\_BLOCKS];

File files[5];

int allocate\_block() {

for (int i = 0; i < MAX\_BLOCKS; i++) {

if (disk[i].next\_block == 0) {

return i;

}

}

return -1;

}

void create\_file(int file\_id) {

int first\_block = allocate\_block();

if (first\_block == -1) {

printf("No blocks available.\n");

return;

}

files[file\_id].first\_block = first\_block;

files[file\_id].last\_block = first\_block;

disk[first\_block].next\_block = -1;

printf("File %d created with block %d\n", file\_id, first\_block);

}

void add\_block(int file\_id) {

int new\_block = allocate\_block();

if (new\_block == -1) {

printf("No blocks available to add.\n");

return;

}

disk[files[file\_id].last\_block].next\_block = new\_block;

files[file\_id].last\_block = new\_block;

disk[new\_block].next\_block = -1;

printf("Block %d added to file %d\n", new\_block, file\_id);

}

void display\_file(int file\_id) {

int block = files[file\_id].first\_block;

while (block != -1) {

printf("%d ", block);

block = disk[block].next\_block;

}

printf("\n");

}

int main() {

for (int i = 0; i < MAX\_BLOCKS; i++) disk[i].next\_block = 0; // Available blocks

create\_file(0);

add\_block(0);

add\_block(0);

display\_file(0);

return 0;

}

OUTPUT:

