

## CSI 402 – Systems Programming

### Updating records in a database (random access) file

#### Handout 2.3

The following program uses the database file (“`emp_db.bin`”) created by the program in Handout 2.2. This program asks user to input information about employees, one at a time, and updates the corresponding record in the database file. The user signals the end of input by giving information for an employee with `id = -1`. For simplicity, the program does not perform any error checking on the input provided by the user.

---

```
#include <stdio.h>
#include <stdlib.h>

#define NEMP 100
#define NAME_MAX 20
#define DB_NAME "emp_db.bin"

struct employee{
    char name[NAME_MAX];
    int empid;
    float salary;
};
typedef struct employee EmpRec;

/* This program asks user to input all the information about */
/* an employee and writes the record for that employee into */
/* the database specified by DB_NAME. */

/* If there is a previous record for the employee, that */
/* record is overwritten. */

/* Recall that all "dummy" records have empid = -1. */

int main(void) {

    FILE *dbf; /* Employee database */
    EmpRec e; /* Temporary record. */
    int i; /* Temporary index. */

    void update_db(FILE *, EmpRec*); /* Prototype */
```

(over)

```

/* Open the database file. We need the mode "r+" to */
/* allow updates.                                     */

if ((dbf = fopen(DB_NAME, "r+")) == NULL) {
    fprintf(stderr, "Could not open file: %s\n", DB_NAME);
    exit(1);
}

/* Obtain employee information and update file until */
/* an employee id of -1 is typed.                     */

/* Employee ids are assumed to be in integers in the range */
/* 0 through NEMP-1.                                       */

/* NOTE: The program does not check for input errors.      */

do {
    printf("Type empid, name and salary: "); fflush(stdout);
    scanf("%d%s%f", &e.empid, e.name, &e.salary);

    if (e.empid != -1) {
        update_db(dbf, &e);
    }
} while (e.empid != -1);

/* All updates have been done. Close the database file. */

if (fclose(dbf) == EOF) {
    fprintf(stderr, "Could not close file: \n", DB_NAME);
}
return 0;
} /* End of main. */

void update_db(FILE *dbf, EmpRec *eptr) {

    /* Updates the database file by writing the information */
    /* given by employee record *eptr.                      */

    /* Reach the appropriate offset in the file number using */
    /* eptr->empid.                                           */

    fseek(dbf, (eptr->empid)*sizeof(EmpRec), SEEK_SET);

    /* Write the information given by the record *eptr.      */
    /* (Previous record, if any, is lost.)                   */

    fwrite((const void *) eptr, sizeof(EmpRec), 1, dbf);

} /* End of update_db */

```