**Технически университет – София**

Курсова работа по

ПИК 2

Изпълнена от:

Студент: Венцислав Дочев

Факултет: ФКСТ

Специалност: КСИ

Група: 35

Факултетен №: 121217092

Проверил:

Десислав Андреев

Съдържание:

1.Задание

2.Функционалност

3.Реализация

4.Употреба

5.Обобщение

1. Задание:

Едно време в дивия северозапад, хората се сбивали най-редовно, но пък за сметка на това, след всеки бой, тези които са били още в съзнание, продължавали да пият и така, ако започнел нов кютек, все едно са били напълно здарви. Това най-често били победителите..., а победените – лежали в безсъзнание някъде по градинките. Направете „турнир“ от биячи в една кръчма, които са записани в книгата на кръчмаря. Зареждайте имената им в контейнер по избор, след което ги сбивайте по двойки. Приемете, че всеки почерпен има по 100% здраве и никой не може да му нанесе повече от 20% щета. Биячите се бият последователно – първо удря единият, после другият – не е честно, ама така е в живия живот. Естествено – бият се докато единият не победи. Генерирайте двубоите чрез rand(). След всеки кръг направете нови двойки от победителите и така по метода на елиминирането в края обявете победител. При началото на всеки бой участнците са с по 100% здраве.

Направете обект „кръчмар“ – той също е бияч по природа, но има 120% здраве и не е в списъка с биячите, а се включва по някое случайно време. Също така при извикване отпечатва „Издислав...“.

Начинът, по който аз виждам как да се реши задачата, е:

Правя списък или масив от структури за биячите.

Пълня го с информация от текестов файл със хедър:

<Години> | <Размер на името> | <Името>

Избирам биячите по id. Като id-то е избрано произволно от функцията rand().

Намирам биячите с избрано id и ги сбивам.

Боя става като се генерира rand() число между 0 и 20.

От здравето на бияча се изважда генерираното число.

Загубилият остава с 0 или по-малко живот.

Загубилия бива изтриван от масива или от списъка от структури.

Масивът се подрежда(ако се работи с масив).

Последният остана елемент от масива или списъка е победителят.

Кръчмаря е отделна структура, която наследява структурата на биячите и я допълва с указател към функция(function pointer).

Отново се генерира rand() число, но този път то решава дали кръчмарят ще участва в битката. Ако не участва структура не се генерира и така не пълни паметта.

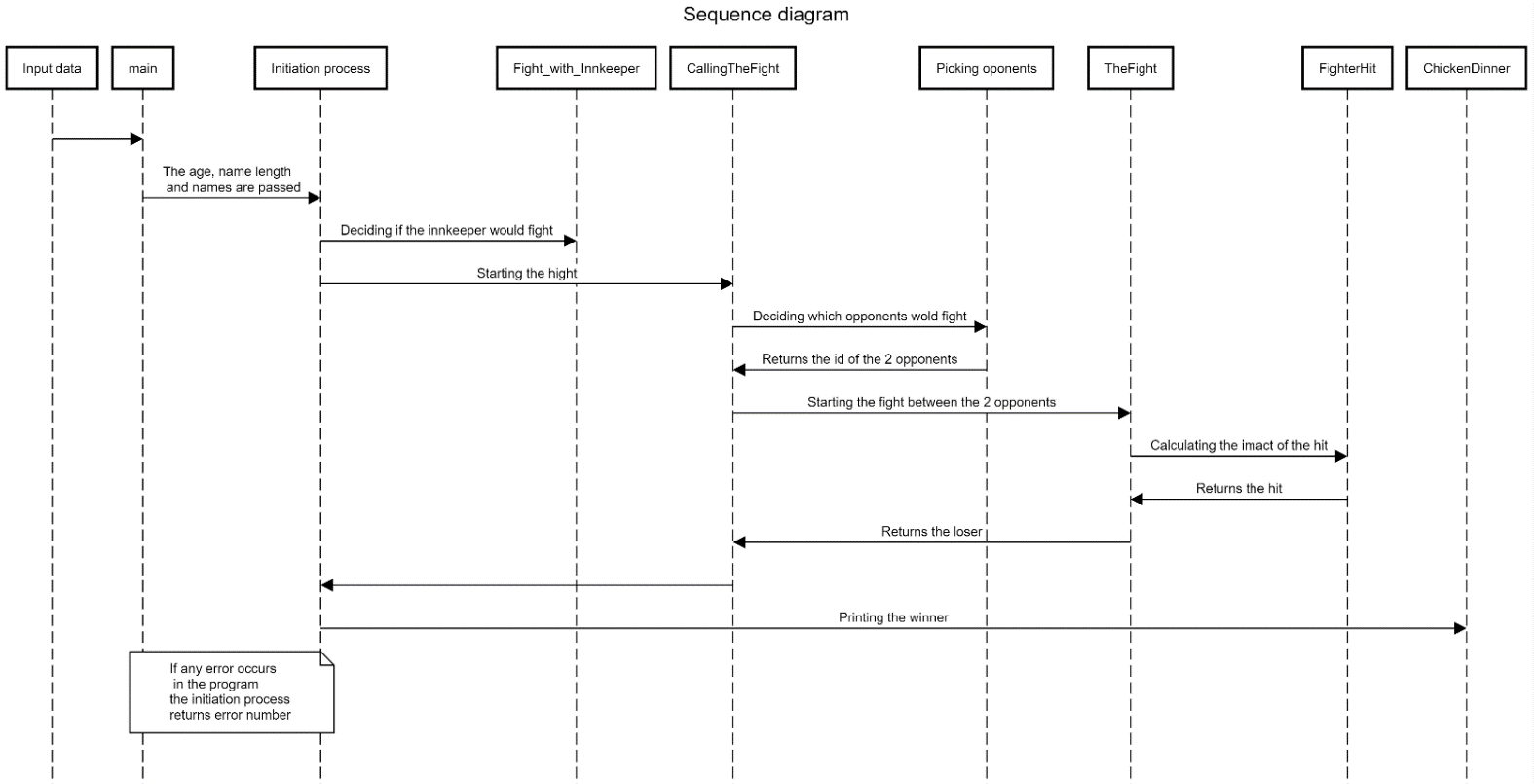
Всеки път, когато е извикан принтира на екран “Izdislav”.

Накрая победителят бива изписван на екрана.

1. Функционалност:

Програмата приема входни данни под формата на списък от бойци.

Връща като отговор победителят от битките.



1. Реализация:

Създаване на хедър файл, сорс файл и основен файл.

Създаване на структурите:

typedef struct fighter

{

uint32\_t id;

uint16\_t age;

size\_t name\_len;

byte \*name;

} ts\_fighter;

typedef struct innkeeper

{

ts\_fighter base\_properties;

int(\*executeFight) (uint32\_t, ts\_fighter \*);

} ts\_innkeeper;

Записване на информацията от файла, подаден чрез конзолата, в структурите.

Годините и размера на името се приемат директно, а за името се прави място в паметта с malloc, а след това се запълва с информация от файла:

if (fscanf(booksource, "%hu | ", &age) != 1) break;

if (NULL == (fighters = (ts\_fighter \*)realloc(fighters, sizeof(ts\_fighter) \* (idx + 1)))) return 3;

fighters[idx].age = age;

if ((fscanf(booksource, "%u | ", &fighters[idx].name\_len)) != 1) return 6;

if (NULL == (fighters[idx].name = (byte \*)malloc(sizeof(byte) \* fighters[idx].name\_len))) return 4;

if (fscanf(booksource, "%s\n", fighters[idx].name) != 1) return 6;

fighters[idx].id = (idx + 1);

Избиране на биячи, чрез генериране на числа, които съответстват на id-тата на бойците.

Функцията приема като информация колко бойци са живи, и връща rand() число:

uint32\_t PickingOponents(uint32\_t numberOpponents)

{

time\_t timer;

uint32\_t generatedNum = 0;

srand((unsigned)time(&timer));

generatedNum = (rand() % numberOpponents);

return generatedNum;

}

Борбата се извършва чрез изваждане на удар от здравето на играча:

uint32\_t TheFight(uint32\_t id1, uint32\_t id2)

{

int fighter1\_HP = FULL\_HP;

int fighter2\_HP = FULL\_HP;

/\* fighter1 hits first so fighter2 takes the damage first \*/

while (1)

{

fighter2\_HP -= FighterHit();

if (fighter2\_HP <= 0) return id2;

fighter1\_HP -= FighterHit();

if (fighter1\_HP <= 0) return id1;

}

}

Генериране на число, което кореспондира на щетата, която даден удар прави:

uint16\_t FighterHit()

{

time\_t clock;

uint16\_t generatedNum = 0;

uint16\_t generatedNum1 = 0;

uint16\_t generatedNum2 = 0;

srand((unsigned)time(&clock));

generatedNum1 = (uint16\_t)(rand() % MAX\_DMG);

generatedNum2 = (uint16\_t)(rand() % MAX\_DMG);

generatedNum = (uint16\_t)((generatedNum1 + generatedNum2) / 2);

return generatedNum;

}

Изчистване на масива от загубилия боец. Подредба на масива:

loser\_number = TheFight(fighter1\_number, fighter2\_number);

for (idx = loser\_number; idx < (number\_fighters - 1); ++idx)

{

fighters[idx] = fighters[idx + 1];

}

Последния оцелял бива принтиран на екрана. Принтирането става като подаване на последния останал елемент от масива на функция, която принтира данните взети от структурата:

void ChickenDinner(ts\_fighter winner)

{

printf("Winner winner chicken dinner\n");

printf("At age of %d, %s won the tournament of the north-western fight club", winner.age, winner.name);

}

Инициализиране на „кръчмаря“ чрез подаване на функция към указателя към функция:

/\* Picking opponent can be used to decice when the innkeeper to show \*/

/\* 100/CHANCE\_OF\_SPAWN is the CHANCE OF SPAWN in percentage if equal to 1 \*/

if (PickingOponents(100 / CHANCE\_OF\_SPAWN) == 1)

{

if (innkeeper\_status == 1) continue;

/\* The innkeeper has shown \*/

printf("IZDISLAV!");

the\_inkeeper.executeFight = Fight\_with\_Innkeeper;

innkeeper\_status = the\_inkeeper.executeFight(number\_fighters, fighters);

number\_fighters--;

}

else

{

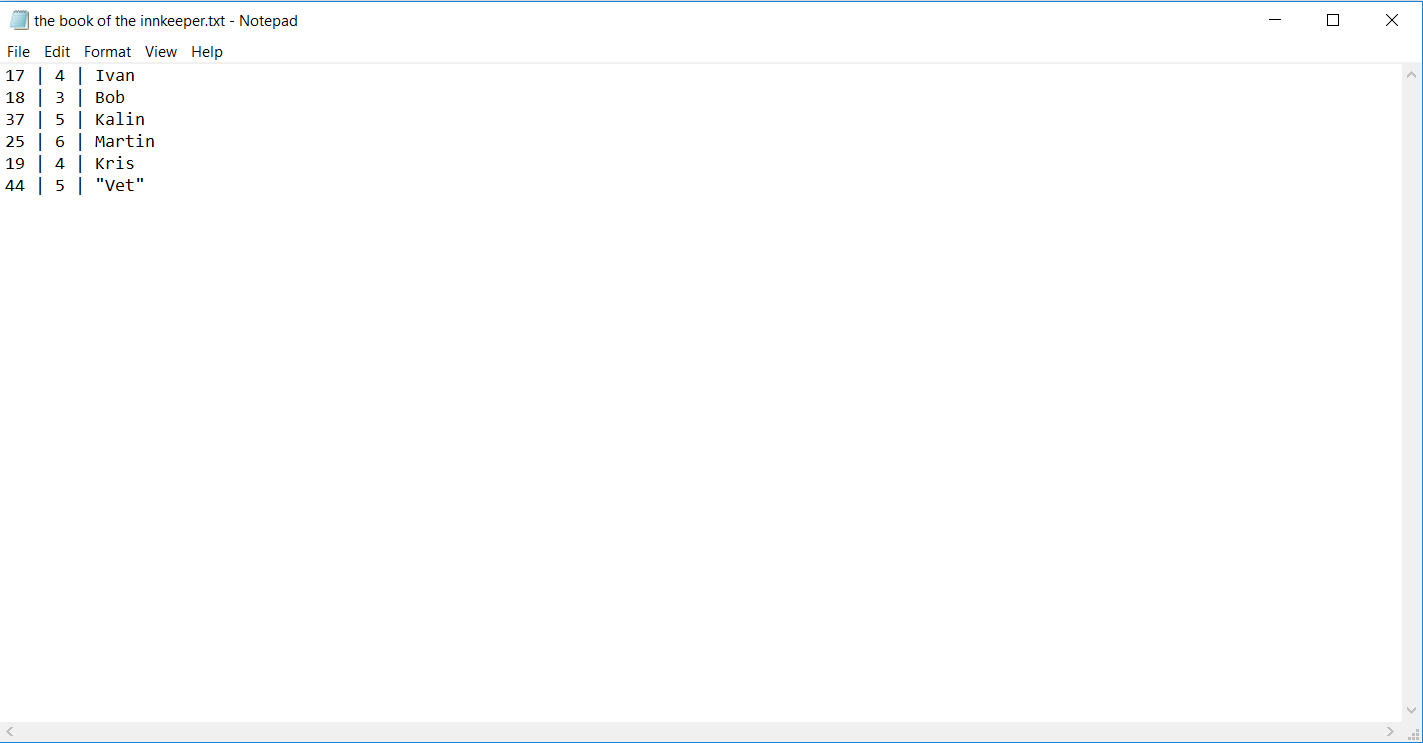
CallingTheFight(fighters, number\_fighters);

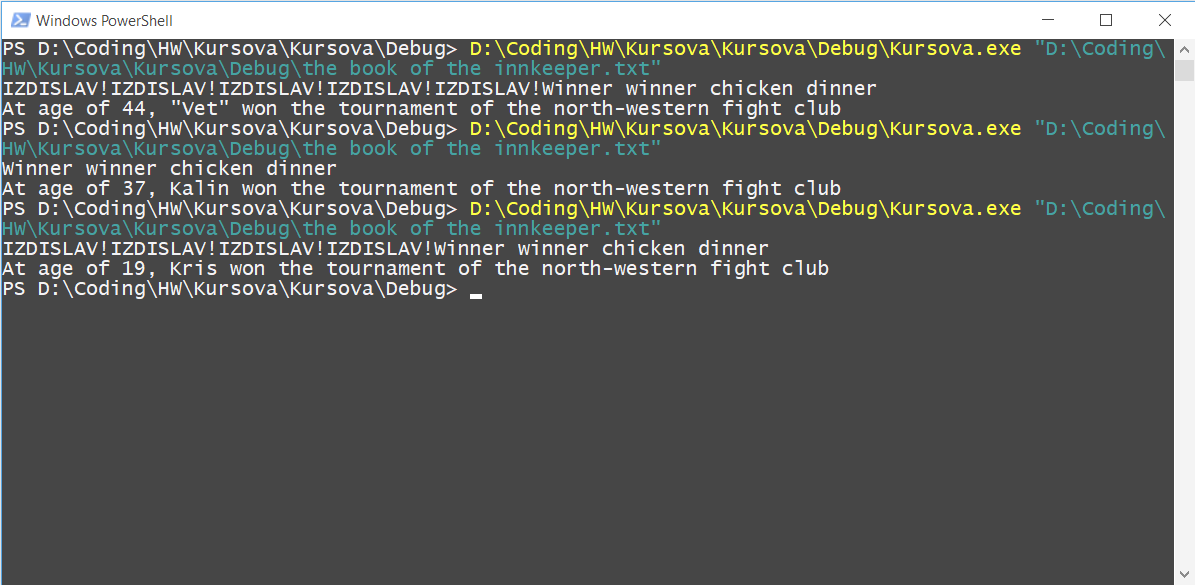
number\_fighters--;

}

1. Употреба:

Входни данни:



Изходни даннни:

1. Обобщение:

Задачата беше от средна трудност за мен. Срещнах трудност при подаването на функция към function pointer в структура, направих малки грешки при четенето от файл. Задачата разви уменията ми при работа с масиви от стуктури и използването на rand() функцията.

1. Source code:

https://github.com/VDochev/Coursework\_2018