

基礎情報演習 1B 第4回～6回レポート

AL16030 笠井信宏

平成 29 年 5 月 30 日

第1章 プログラム解説

1.1 プログラム記載

quiz.c, quiz.h, typing.c, typing.h, menu.c, menu.h をそれぞれ載せる.

Listing 1.1: quiz.c

```
1 #include <stdlib.h>
2 #include "quiz.h"
3 #define N 100
4
5 int quizReader(QUIZ *quiz,int *n){
6     int i = 0;
7     FILE *fp;
8     fp = fopen("quiz.txt","r");
9     if(fp == NULL){
10         printf("ファイルの読み込みに失敗しました\n");
11         return -1;
12     }
13     while(fscanf(fp,"%[^,],%[^,],%[^,],%d",quiz[i].question,quiz[i].ans1,quiz[i].
        ans2,&quiz[i].correct_ans) != EOF && i < N){
14         i++;
15     }
16     *n = i-1;
17     return 0;
18 }
19
20 void input(){
21     QUIZ *quizzes = (QUIZ *)malloc(sizeof(QUIZ)*100);
22     int num,i,ans,score = 0;
23     if(quizReader(quizzes,&num)){
24         printf("エラーが発生したようです\n");
25         return;
26     }
27     for(i=0;i<num;i++){
28         printf(quizzes[i].question);
29         printf("\n1:□%s,2:□%s\n",quizzes[i].ans1,quizzes[i].ans2);
30         scanf("%d",&ans);
31         if(ans == quizzes[i].correct_ans){
32             printf("CORRECT!\n");
33             score += 5;
34         } else {
35             printf("WRONG!\n");
36         }
37         do {
38             ans = getchar();
39         } while(ans != '\n' && ans != EOF);
```

```

40     printf("score: %d\n",score);
41 }
42 }

```

u.h をそれぞれ載せる.

Listing 1.2: quiz.h

```

1  #ifndef QUIZ_H_
2  #define QUIZ_H_
3
4  #include <stdio.h>
5
6  struct _quiz {
7      char question[50];
8      char ans1[50];
9      char ans2[50];
10     int correct_ans;
11 };
12
13 typedef struct _quiz QUIZ;
14
15 int quizReader(QUIZ *quiz, int *n);
16 void input();
17
18 #endif

```

Listing 1.3: typing.c

```

1  #include <stdio.h>
2  #include <string.h>
3  #include <stdlib.h>
4  #include <time.h>
5  #include "typing.h"
6  #define MAXLEN 100
7
8  int DataReader(DATA *dataArray, int *n){
9      FILE *fp;
10     int i = 0;
11     fp = fopen("typing.txt", "r");
12     if(fp == NULL) return -1;
13     while(fscanf(fp, "%[^,], %d\n", dataArray[i].string, &dataArray[i].score) !=
14           EOF || i > MAXLEN){
15         i++;
16     }
17     *n = i - 1;
18     return 0;
19 }
20
21 int typing(DATA *dataArray, int n){
22     double diff;
23     double remain_time = 20.0;
24     time_t starttime = time(NULL), endtime;
25     int i=1, number, score = 0;
26
27     char ans[50];
28     while(1){

```

```

28     number = rand()%n;
29     printf("%s\n",dataArray[number].string);
30     // 計測
31     starttime = time(NULL);
32     scanf("%s",ans);
33     endtime = time(NULL);
34
35     fflush(stdin);
36
37     if(strcmp(dataArray[number].string,ans) == 0){
38         printf("Great!\n");
39         score += dataArray[number].score;
40         remain_time += (double)dataArray[number].score;
41     } else {
42         printf("Bad...\n");
43         remain_time -= dataArray[number].score;
44     }
45     remain_time -= difftime(endtime,starttime);
46     if(remain_time > 0){
47         printf("%fsecond left. Your current score is %d\n",remain_time,
48             score);
49     } else {
49         printf("TIME UP! Game Over...\n");
50         break;
51     }
52 }
53 return score;
54 }

```

Listing 1.4: typing.h

```

1  #ifndef TYPING_H_
2  #define TYPING_H_
3
4  #define MAXLEN 100
5
6  struct _data{
7      char string[50];
8      int score;
9  };
10
11 typedef struct _data DATA;
12
13 int DataReader(DATA *dataArray, int *n);
14 int typing(DATA *dataArray, int n);
15
16 #endif /* TYPING_H_ */

```

Listing 1.5: menu.c

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include "menu.h"
4  #define N 100
5
6  void clean_stdio(){
7      int ch;

```

```

8   do {
9       ch = getchar();
10  } while(ch != '\n' && ch != EOF );
11  }
12
13 void menu(){
14     DATA *dataArray = (DATA *)malloc(sizeof(DATA)*N);
15     int n,success,ret = 1;
16     char ch;
17     while(ret==1){
18         printf("----menu---\nq: Quiz\n\t: Typing\n\t: Exit\n>");
19         ch = getchar();
20         switch(ch){
21             case 'q':
22                 input();
23                 break;
24             case 't':
25                 success = DataReader(dataArray,&n);
26                 if(!success) typing(dataArray,n);
27                 break;
28             case 'e':
29                 ret = 0;
30                 break;
31             case '\n':
32                 break;
33             default:
34                 printf("invalid command\n");
35                 clean_stdio();
36                 break;
37         }
38     }
39     free(dataArray);
40 }

```

Listing 1.6: menu.h

```

1 #ifndef MENU_H_
2 #define MENU_H_
3 #include "quiz.h"
4 #include "typing.h"
5
6 void clean_stdio();
7 void menu();
8 #endif

```

出力記載

次に，出力を記載する．

Listing 1.7: ”出力”

```

1 $.\menu.exe
2 ----menu----
3 q: Quiz

```

```

4 t: Typing
5 e: Exit
6 >q
7 What_is_the_first_month_in_a_year?
8 1: January,2: December
9 1
10 CORRECT!
11 score: 5
12
13 Where_is_the_capital_of_Japan?
14 1: Kyoto,2: Tokyo
15 1
16 WRONG!
17 score: 5
18
19 What_is_the_name_of_this_building?
20 1: Building_2,2: Building_4
21 2
22 WRONG!
23 score: 5
24 ---menu---
25 q: Quiz
26 t: Typing
27 e: Exit
28 >t
29 AppEngine
30 AppEngine
31 Great!
32 22.000000 second left. Your current score is 4
33 AppEngine
34 AppEngine
35 Great!
36 22.000000 second left. Your current score is 8
37 NoSQL
38 NoSQL
39 Great!
40 22.000000 second left. Your current score is 11
41 NoSQL
42 NoSQL
43 Great!
44 22.000000 second left. Your current score is 14
45 AppEngine
46 AppEngine
47 Great!
48 23.000000 second left. Your current score is 18
49 NoSQL
50 NoSQL
51 Great!
52 20.000000 second left. Your current score is 21
53 MySQL
54 MYSQL
55 Bad...
56 12.000000 second left. Your current score is 21
57 MySQL
58 a
59 Bad...
60 4.000000 second left. Your current score is 21

```

```
61 NoSQL
62 a
63 Bad...
64 TIME UP! Game Over...
65 ----menu----
66 q: Quiz
67 t: Typing
68 e: Exit
69 >e
```

1.2 考察

今回難点であったのは標準入力のリセットであった。 `getchar` 関数を利用する部分において、予期せぬ入力があったときに標準入力にバッファが残ってしまい、それがプログラムに予期せぬ動作を起こしていた。それを解決するために利用した `fflush` 関数が、環境事に実装が違うためにエラーが発生していたため、 `clean_stdio` 関数を実装することで解決した。 Windows での実行では `fflush` で問題なかったが、Linux では標準入力にテキストが残ってしまっているようだった。