

付録4にヒントがあるので適宜参照すること。

1 語句に関する問題

空欄 (1)～(10) に最適な言葉を語群から記号で答えなさい。また、空欄 (A)～(E) に最適な言葉を a または b の記号で答えなさい。(1 点 × 15 問 = 15 点)

環境変数は、通常プログラムが起動される際にシェルからプログラムに (1) される。環境変数を表示する (2) コマンドは、シェルの外部コマンドとして実装することが (A) である。シェルの環境変数を書換えるコマンドは、シェルの外部コマンドとして実装することが (B) である。

環境変数を変更した上でプログラムを実行する env コマンドは、シェルの外部コマンドとして実装することが (C) である。env コマンドは fork システムコールを使用する (D)。

C 言語からシェルの文法でコマンドを実行することができる (3) 関数は、関数の内部で fork システムコールを使用する (E)。(3) 関数は渡された文字列を解釈・実行するために、(4) を使用する。

プログラムを新しいプロセスで実行するための方式は、プロセスを作るシステムコールとプログラムを実行するシステムコールの二つのシステムコールを使用する (5) 方式と、一つのシステムコールで行う (6) 方式がある。

シェルが入出力のリダイレクトをするときは、fork した (7) の標準入出力を操作した上で、(7) が目的のプログラムを exec する。実行されるプログラムにリダイレクト機能が備わっている (F)。リダイレクトは、まず、標準入出力を (8) し、次に、リダイレクト先ファイルを (9) することでなされる。

(A),(B),(C) の候補：(a) 可能, (b) 不可能

(D),(E),(F) の候補：(a) 必要がある, (b) 必要はない

語群：(あ)close, (い)open, (う)printenv,
(え)spawn, (お)system, (か)fork-exec,
(き)コピー, (く)子プロセス, (け)/bin/sh

(1)	(き)	(2)	(う)	(A)	(a)	(B)	(b)
(C)	(a)	(D)	(b)	(3)	(お)	(E)	(a)
(4)	(け)	(5)	(か)	(6)	(え)	(7)	(く)
(F)	(b)	(8)	(あ)	(9)	(い)		

2 Cプログラムの実行結果

1. 付録1の p0.c の実行結果を答えなさい。(5 点)

B

2. 付録1の p1.c の実行結果を答えなさい。(5 点)

X=Y

A=B

B=C

3. 付録1の p2.c の実行結果を答えなさい。(5 点)

20

10

4. 付録1の p3.c の実行結果を答えなさい。(5 点)

child

parent

finish

5. 付録1の p4.c の実行結果を答えなさい。(5 点)

child

aaa

parent

aaa

6. 付録1の p5.c の実行結果を答えなさい。(5 点)

0

1

1

システムプログラミングⅡ 令和5年度 前期末試験

(2023.08.02 重村 哲至)

IE4

番 氏名

模範解答

3 myshell プログラム

付録2のC言語プログラムは、授業の課題12のプログラムを作成途中のものです。このプログラムと付録3の実行結果について答えなさい。

- (1) プログラム中の空欄 (###(A)###) に適切な記述を答えなさい。 (4点)

```
setenv(args[1], args[2], 1)<0
```

- (2) プログラム中の空欄 (###(B)###) に適切な記述を答えなさい。 (4点)

```
unsetenv(args[1])<0
```

- (3) プログラム中の空欄 (###(C)###) に適切な1行の記述を答えなさい。 (4点)

```
close(fd);
```

- (4) プログラム中の空欄 (###(D)###) に適切な1行の記述を答えなさい。 (4点)

```
int nfd = open(path, flag, 0644);
```

- (5) プログラム中の空欄 (###(E)###) に適切な1行の記述を答えなさい。 (4点)

```
exit(1);
```

- (6) プログラム中の空欄 (###(F)###) に適切な記述を答えなさい。 (4点)

```
pid==0
```

- (7) プログラム中の空欄 (###(G)###) に適切な1行の記述を答えなさい。 (4点)

```
exit(1);
```

- (8) プログラム中の空欄 (###(H)###) に適切な1行の記述を答えなさい。 (4点)

```
externalCom(args);
```

- (9) 入力用に redirect 関数を呼び出すために追加するプログラムと、追加する位置 (行) を答えなさい。 (4点)

54 行の直前に

```
if (ifile!=NULL) {  
    redirect(0, ifile, O_RDONLY);  
}
```

- (10) 入力用に redirect 関数を呼び出すために追加するプログラムと、追加する位置 (行) を答えなさい。 (4点)

54 行の直前に

```
if (ofile!=NULL) {  
    redirect(1, ofile,  
            O_WRONLY|O_TRUNC|O_CREAT);  
}
```

- (11) 実行結果 6-9 行は環境変数を作ることができるかテストしています。9行の表示を答えなさい。 (4点)

C

- (12) 実行結果 8-12 行は環境変数を上書きできるかテストしています。12行の表示を答えなさい。 (4点)

ja_JP

- (13) 実行結果 15-22 行は何のテストか答えなさい。 (4点)

出力リダイレクトが、ファイルが存在しないときでも正しく動作すること。入力リダイレクトが正しく動作すること。

- (14) 実行結果 26,27 行は、出力リダイレクトのエラー処理をテストするつもりでした。しかし、適切ではありませんでした。理由を簡潔に説明しなさい。 (3点)

出力リダイレクトではなく、プログラム prog の実行 (exec) でエラーが発生しているので、リダイレクト機能のテストになっていない。

システムプログラミングⅡ 令和5年度 前期末試験

(2023.08.02 重村 哲至)

IE4

番 氏名

模範解答

付録1：テストプログラム

```
// p0.c
#include <stdio.h>
#include <unistd.h>
char *args[]={"printenv", "A", NULL};
char *envs[]={"X=Y", "A=B", "B=C", NULL};
int main(int argc, char *argv[]) {
    execve("/usr/bin/printenv", args, envs);
    printf("finish\n");
    return 0;
}
```

```
// p1.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
char *args[]={"printenv", NULL};
char *envs[]={"X=Y", "A=B", "B=C", NULL};
int main(int argc, char *argv[]) {
    unsetenv("X");
    execve("/usr/bin/printenv", args, envs);
    printf("finish\n");
    return 0;
}
```

```
// p2.c
#include <stdio.h>
#include <stdlib.h>
#include <sys/wait.h>
#include <unistd.h>
int main(int argc, char *argv[]) {
    int a = 10;
    if (fork()==0) {
        a = 20;
        printf("%d\n", a);
        exit(0);
    } else {
        int stat;
        wait(&stat);
        printf("%d\n", a);
    }
    return 0;
}
```

```
// p3.c
#include <stdio.h>
#include <stdlib.h>
```

```
#include <sys/wait.h>
#include <unistd.h>
int main(int argc, char *argv[]) {
    if (fork()==0) {
        printf("child\n");
        exit(0);
        printf("exit\n");
    } else {
        int stat;
        wait(&stat);
        printf("parent\n");
    }
    printf("finish\n");
    return 0;
}
```

```
// p4.c
#include <stdio.h>
#include <sys/wait.h>
#include <unistd.h>
int main(int argc, char *argv[]) {
    if (fork()==0) {
        printf("child\n");
    } else {
        int stat;
        wait(&stat);
        printf("parent\n");
    }
    execlp("echo", "echo", "aaa", NULL);
    printf("exit\n");
    return 0;
}
```

```
// p5.c
#include <stdio.h>
#include <sys/wait.h>
#include <unistd.h>
int main(int argc, char *argv[]) {
    for (int i=0; i<2; i++) {
        if (fork()==0) {
            printf("%d\n", i);
        } else {
            int stat;
            wait(&stat);
        }
    }
    return 0;
}
```

システムプログラミングⅡ 令和5年度 前期末試験

(2023.08.02 重村 哲至)

IE4

番 氏名

模範解答

付録2：myshell プログラム

```
1 #include <stdio.h>
2 ... include 省略 ...
3 #define MAXLINE 1000
4 #define MAXARGS 60
5 int parse(char *p, char *args[]) { ... 省略 ... }
6 void cdCom(char *args[]) { ... 省略 ... }
7 void setenvCom(char *args[]) {
8     if (args[1]==NULL || args[2]==NULL ||
9         args[3]!=NULL) {
10         fprintf(stderr,"Usage: setenv NAME VAL\n");
11     } else if (###(A)###) {
12         perror(args[1]);
13     }
14 void unsetenvCom(char *args[]) {
15     if (args[1]==NULL || args[2]!=NULL) {
16         fprintf(stderr,"Usage: unsetenv NAME\n");
17     } else if (###(B)###) {
18         perror(args[1]);
19     }
20 }
21 char *ofile;
22 char *ifile;
23 void findRedirect(char *args[]) {
24     int i, j;
25     ofile = ifile = NULL;
26     for (i=j=0; args[i]!=NULL; i++) {
27         if (strcmp(args[i], "<")==0) {
28             ifile = args[++i];
29             if (ifile==NULL) break;
30         } else if (strcmp(args[i], ">")==0) {
31             ofile = args[++i];
32             if (ofile==NULL) break;
33         } else {
34             args[j++] = args[i];
35         }
36     }
37     args[j] = NULL;
38 }
39 void redirect(int fd,char *path,int flag) {
40     ###(C)###
41     ###(D)###
42     if (nfd<0) { // 他のエラーは起こらないはず
43         perror(path);
44         ###(E)###
45     }
46 }
```

```
47 void externalCom(char *args[]) {
48     int pid, status;
49     if ((pid = fork()) < 0) {
50         perror("fork");
51         exit(1);
52     }
53     if (###(F)###) {
54         execvp(args[0], args);
55         perror(args[0]);
56         ###(G)###
57     } else {
58         while (wait(&status) != pid)
59             ;
60     }
61 }
62 void execute(char *args[]) {
63     if (strcmp(args[0], "cd")==0) {
64         cdCom(args);
65     } else if (strcmp(args[0], "setenv")==0) {
66         setenvCom(args);
67     } else if (strcmp(args[0], "unsetenv")==0) {
68         unsetenvCom(args);
69     } else {
70         ###(H)###
71     }
72 }
73 int main() {
74     char buf[MAXLINE+2];
75     char *args[MAXARGS+1];
76     for (;;) {
77         printf("Command: ");
78         if (fgets(buf,MAXLINE+2,stdin)==NULL) {
79             printf("\n");
80             break;
81         }
82         if (strchr(buf, '\n')==NULL) {
83             fprintf(stderr, "行が長すぎる\n");
84             return 1;
85         }
86         if (!parse(buf,args)) {
87             fprintf(stderr, "引数が多すぎる\n");
88             continue;
89         }
90         findRedirect(args);
91         if (args[0]!=NULL) execute(args);
92     }
93     return 0;
94 }
```

システムプログラミングⅡ 令和5年度 前期末試験

(2023.08.02 重村 哲至)

IE4

番

氏名

模範解答

付録3：myshell の実行結果

```
1 % make
2 cc -Wall -std=c99 -o myshell myshell.c
3 sigemura@micro MyShell % ./myshell
4 Command: printenv LANG
5 ja_JP.UTF-8
6 Command: printenv LC_TIME
7 Command: setenv LC_TIME C
8 Command: printenv LC_TIME
9 ### 何か表示される ###
10 Command: setenv LC_TIME ja_JP
11 Command: printenv LC_TIME
12 ### 何か表示される ###
13 Command: unsetenv LC_TIME
14 Command: printenv LC_TIME
15 Command: ls
16 Makefile      myshell      myshell.c
17 Command: ls > a.txt
18 Command: cat < a.txt
19 Makefile
20 a.txt
21 myshell
22 myshell.c
23 Command: echo abcdefgh > a.txt
24 Command: cat a.txt
25 abcdefgh
26 Command: ./prog abcdefgh > b.txt
27 ./prog: No such file or directory
28 Command: chmod 000 a.txt
29 Command: echo 12345678 > a.txt
30 a.txt: Permission denied
31 Command: cat < a.txt
32 a.txt: Permission denied
33 Command:
```

付録4：ヒント

環境変数に関するコマンド

```
printenv [name]
env name1=value1 name2=value2 ... command
```

C 言語の関数・変数やシステムコール

```
// ファイル関係
int open(char *path, int oflag, ...);
oflag:
    O_RDONLY (読み込み)
    O_WRONLY (書き込み)
    O_CREAT (ファイル作成)
    O_TRUNC (ファイル切詰め)

int close(int fd);

// 環境変数に関する関数・変数
extern char **environ;
char *getenv(char *name);
int setenv(char *name, char *val, int overwrite);
int putenv(char *string);
int unsetenv(char *name);

// プログラム実行関係
int execve(char *path,
            char *argv[], char *envp[]);
int execvp(char *file,
            char *argv[], char *envp[]);
int execl(char *path, char *argv0,
            *argv1, ... , argvn, NULL);
int execlp(char *file, char *argv0,
            *argv1, ... , argvn, NULL);

// プロセス生成・終了関係
int fork(void);
void exit(int status);
pid_t wait(int *status);

// その他関数
int system(char *command);
```