システムプログラミング H27 年度 前期期末試験

(2015.08.04 重村 哲至)

IE4 ____ 番 氏名 模範解答

1. 次の Unicode を UTF-8 エンコーディングしたバイト列を 16 進数で答えなさい。(5 点× 3 問=15 点)

u+0030(半角‘0’): 0x30

u+00a5(半角‘¥’): 0xc2 0xa5

u+4e9c(漢字‘亜’): 0xe4 0xba 0x9c

2. 別紙 1 の C 言語プログラム (exa.c, exb.c) について答えなさい。(5 点× 4 問=20 点)

(1) exa.c は実行例のように表示の言語をコマンド行で指定することができる date コマンドとして動作します。プログラム中の空欄 (a),(b) に補うべきプログラムを以下に書きなさい。

```
$ exa
Sat Aug  1 18:37:32 JST 2015
$ exa ja_JP.UTF-8
2015 年 8 月 1 日 土曜日 18 時 37 分 29 秒 JST
```

(a)

```
setenv("LC_TIME", argv[1], 1)
```

(b)

```
execlp("date", "date", NULL);
```

(2) exb.c は実行例のようにコマンド行の第 1 引数で指定したディレクトリに移動して、コマンド行の残りの引数で示されるコマンドを実行します。プログラム中の空欄 (a),(b) に補うべきプログラムを以下に書きなさい。

```
$ mkdir A
$ echo aaa > A/a
$ exb A cat a
aaa
```

(a)

```
chdir(argv[1])
```

(b)

```
execvp(argv[2], &argv[2]);
```

3. 別紙 1 の C 言語プログラム (exc.c) は、任意の環境変数を設定して date コマンドを実行します。次の問に答えなさい。(10 点× 2 問=20 点)

(1) 次の実行例になるようにプログラム中の空欄に補うべき記述を答えなさい。

```
$ exc LC_TIME=ja_JP.UTF-8 TZ=GMT
2015 年 8 月 1 日 土曜日 10 時 37 分 13 秒 JST
2015 年 8 月 1 日 土曜日 01 時 37 分 13 秒 GMT
```

```
putenv(argv[i]);
pid=fork();
if (pid==0) {
  execlp("date", "date", NULL);
  exit(1);
}
```

(2) 次の実行例になるようにプログラム中の空欄に補うべき記述を答えなさい。

```
$ exc LC_TIME=ja_JP.UTF-8 TZ=GMT
2015 年 8 月 1 日 土曜日 10 時 37 分 07 秒 JST
Sat Aug  1 01:37:07 GMT 2015
```

```
pid=fork();
if (pid==0) {
  putenv(argv[i]);
  execlp("date", "date", NULL);
  exit(1);
}
```

4. 別紙 2 の C 言語プログラム (myshell3.c) は、授業で紹介した myshell2.c を改良したものです。プログラムをよく読んで以下の問に答えなさい。

(1) 次の実行例では入力が下線部 4 箇所で空白になっています。入力を実行例中に書き込みなさい。(5 点× 4 問=20 点)

```
$ myshell3
Command:  printenv LANG
ja_JP.UTF-8
Command:  set LANG C
Command:  printenv LANG
C
Command:  unset LANG
Command:  printenv LANG
Command:  pwd
/Users/sigemura
Command:  cd ..
Command:  pwd
/Users
Command:  cd
cd の引数が不足
Command:
```

(2) 次の実行例のように、引数なしの cd コマンドがホームディレクトリへカレントディレクトリを移すことにします。実行例をよく見て問に答えなさい。(5 点× 2 問=10 点)

```
Command: cd /tmp
Command: pwd
/tmp
Command: cd
Command: pwd
/Users/sigemura
Command: printenv HOME
/Users/sigemura
Command: set HOME /Users/sigemura/bin
Command: cd
Command: pwd
/Users/sigemura/bin
Command: unset HOME
Command: cd
HOME がみつからない
Command:
```

a. myshell は、ホームディレクトリのパスを何から知りますか。

HOME 環境変数の値をホームディレクトリのパスとして認識している。

b. C 言語プログラムで環境変数の値を知るために使用する関数の名前を書きなさい。

getenv 関数

c. 実行例のように cd コマンドの機能を拡張子した myshell3 の cd() 関数を書きなさい。(20 点)

```
void cd(char *args[]) {
  char *dir = args[1];
  if (dir==NULL) {
    dir = getenv("HOME");
    if (dir==NULL) {
      fprintf(stderr,"HOME がみつからない\n");
      return;
    }
  }
  if (chdir(dir)<0)
    perror(dir);
}
```

注意：出題の都合上、エラー処理が省略してあることがある。

```
// exa.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main(int argc, char *argv[]) {
  if (argc>1) {
    if (/* (a) */!=0) {
      perror("setenv");
      exit(1);
    }
  }
  /* (b) */
  perror("date");
  exit(1);
}
```

```
// exb.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main(int argc, char *argv[]) {
  if (argc<=2) {
    fprintf(stderr,
    "Usage : %s dir command\n",argv[0]);
    exit(1);
  }
  if (/* (a) */!=0) {
    perror(argv[1]);
    exit(1);
  }
  /* (b) */
  perror(argv[2]);
  exit(1);
}
```

```
// exc.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main(int argc, char *argv[]) {
  for (int i=1; i<argc; i++) {
    int pid,stat;
    /*
     *  ここにプログラムを補う
     */
    while(pid==wait(&stat))
      ;
  }
  exit(0);
}
```

```
// myshell3.c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>

void parse(char *buf, char *args[]) {
  while (*buf!='\0') {
    while (*buf==' ' || *buf=='\t')
      *buf++ = '\0';
    if (*buf=='\0') break;
    *args++ = buf;
    while (*buf!='\0' && *buf!=' ' && *buf!='\t')
      buf++;
  }
  *args = NULL;
}

void cd(char *args[]) {
  if (args[1]==NULL)
    fprintf(stderr,"cd の引数が不足\n");
  else if (chdir(args[1])<0)
    perror(args[1]);
}

void set(char *args[]) {
  if (args[1]==NULL||args[2]==NULL)
    fprintf(stderr,"set の引数が不足\n");
  else if (setenv(args[1],args[2],1)<0)
    perror(args[1]);
}

void unset(char *args[]) {
  if (args[1]==NULL)
    fprintf(stderr,"unset の引数が不足\n");
  else if (unsetenv(args[1])<0)
    perror(args[1]);
}
```

```
void execute(char *args[]) {
  if (strcmp(args[0],"cd")==0) {
    cd(args);
  } else if (strcmp(args[0],"set")==0) {
    set(args);
  } else if (strcmp(args[0],"unset")==0) {
    unset(args);
  } else {
    int pid, status;
    if ((pid = fork()) < 0) {
      perror("fork");
      exit(1);
    }
    if (pid==0) {
      execvp(*args, args);
      perror(*args);
      exit(1);
    }
    while (wait(&status) != pid)
      ;
  }
}

int main() {
  char buf[1024];
  char *args[64];
  char *p;
  for (;;) {
    printf("Command: ");
    if (fgets(buf,1024,stdin)==NULL) {
      printf("\n");
      exit(0);
    }
    if ((p=index(buf, '\n'))!=NULL)
      *p = '\0';
    parse(buf,args);
    if (args[0]!=NULL) execute(args);
  }
}
```