

プログラミング応用 前期中間試験

※提出する全てのファイル内に、コメントとして自分の番号と名前を書くこと。

準備 この試験で使用するテンプレートファイルが事前に用意されている。端末内で、以下のコマンドを実行して、テンプレートファイルが入ったフォルダをコピーしておくこと。

```
$ cp -r /usr/local/common/kogai/ouyou201806 . (←ここにピリオド)
$ cd ouyou201806 (コピーしたフォルダに移動する)
$ ls (フォルダ内のファイルを確認すると、以下の 6 個のファイルがコピーされている)
cat.txt mycmd.sh mysh.sh
in.txt mygit.sh mysys.c
```

1 シェルスクリプトを用いて、次の処理をするプログラムを「mysh.sh」という名前のファイルで作成しなさい。

1. 変数 num1 に 35 を代入する
2. 変数 num2 に 10 を代入する
3. 変数 num3 に、num1 から num2 を割った余りを代入する
4. 変数 num3 の値を標準出力に出力する

[実行結果]

5

2 シェルスクリプトを用いて、「与えられた値リストの中から最大値を見つけて、標準出力に出力する」プログラムを「mysh.sh」という名前のファイルに作成しなさい。以下のシェルスクリプトの空欄部分に適切な処理を追加して完成させなさい。

```
max=0
for i in 5 9 1 2 10 4 6 7
do
    #
    #ここに適切な処理を追加する（変数 max より大きい値が見つかったら、max にその値を代入する）
    #
done
echo ${max}
```

[実行結果（問 1 の続きにプログラムを書いた場合）]

5 (←問 1 の結果)
10 (←問 2 の結果)

3 コマンドライン引数として 1 つファイル名を受け取り、そのファイルの中身を読み込み、標準出力に出力するプログラムを以下の処理の流れとなるように作成する。

1. コマンドライン引数として 1 つファイル名を受け取り、そのファイル名を `printf()` で表示する
2. システムコールを使って、受け取ったファイル名を読み込み専用で開く
3. システムコールを使って、ファイルから文字列を読み込む
4. システムコールを使って、読み込んだ文字列を標準出力に出力する
5. システムコールを使って、ファイルを閉じる

この処理をするプログラムを、以下に示す C プログラムの空欄部分に適切な処理を追加して、「mysys.c」という名前のファイルに完成させなさい。

```
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>
#include <string.h>

int main(int argc, char* argv[])
{
    char s[256];
    //
    // ここに適切な処理を追加する
    //
    return (0);
}
```

in.txt (読み込むファイルの例)

Good Morning!

[実行結果 (コマンドライン引数に「in.txt」を指定して実行した場合)]

```
$ ./a.out in.txt
コマンドライン第 1 引数 : in.txt
Good Morning!
```

4 GitHub を使ってソースコードのバージョン管理をする場合を考える。以下の (1) と (2) に示す流れの作業をするためには、それぞれどのような UNIX コマンドを実行すればよいのか「mygit.sh」という名前のファイルに記述して答えなさい。ただし、演習室の PC では動作確認できないことに注意すること。

(1) ファイルの作成, ステージング, コミット, プッシュについて

1. 「sort.c」という名前の空のファイルを作る
2. 全てのファイルをステージングする
3. 「ソーティング追加」というコメントで、変更の記録をコミットする
4. master ブランチのコミット内容を GitHub にプッシュする

(2) ブランチの作成, ブランチの移動について

1. 「quicksort」という名前のブランチを作成する
2. 「quicksort」ブランチに移動する

5 UNIX のコマンドを使って、以下の (1) ～ (3) の作業をすることを考える。それぞれどのような UNIX コマンドを実行すればよいのか「mycmd.sh」という名前のファイルに記述して答えなさい。

(1) プロセス管理とリダイレクトについて

「ps u」コマンドの実行結果を「psresult.txt」というファイルに保存する

(2) 文字列の出力について

「知的財産権」について説明した以下の文章の 空欄 に当てはまる文字列を、標準出力（端末）に出力する

「思想又は感情を創作的に表現したものであり、文芸、学術、美術又は音楽の範囲に属するもの」を 空欄 といい、オリジナリティのあるプログラムコードもこれに含まれる。
--

(3) パイプ機能について

「吾輩は猫である」の全文が保存されている「cat.txt」という名前のテキストファイルに対して、「である。」を含む行のみを標準出力に出力する（句点も含むことに注意する）

(各 20 点)

問題はここまで

定期試験の実施について

試験中に使用できるもの

- 筆記用具（メモ用紙は必要な人に配布）
- 演習室のコンピュータ一台（一つの机に一人の配置で、座る場所はどこでもよい）

試験中に参照できるもの

- 自分のホームディレクトリ（ホームフォルダ）以下に保存されているファイル
(定期試験では紙媒体のものは参照不可)
- * 上記以外の情報を参照することは不正行為とする
(例：USB で接続された機器に保存されているファイルの参照など)
- * 試験中は、演習室外へのネットワークアクセスは遮断される

答案の提出

- 提出する全てのファイル内に、自分の学科の出席番号と氏名をコメントとして書く
- 保存したファイルは次のように「report」コマンドで提出する
(ちゃんと提出できた場合は、「Succeed.」と画面に表示される)
\$ ~kogai/report ouyou1mid 「プログラムファイル」
- 複数のファイルを提出する場合は、report コマンドを分けて提出する

```
$ ~kogai/report ouyou1mid mysh.sh
$ ~kogai/report ouyou1mid mysys.c
$ ~kogai/report ouyou1mid mygit.sh
$ ~kogai/report ouyou1mid mycmd.sh
```

- 同じ問題に対して、複数の提出ファイルが存在した場合は、更新日時が新しい方を提出ファイルとする

前期中間試験 模範解答 (平均 88.6 点)

採点について コンパイル時にエラーとなる箇所は -4 点, 実行可能だが処理内容が問題の意図と違う箇所は -2 点を基本とする。

問 1

```
num1=35
num2=10
num3='expr ${num1} % ${num2}'
echo ${num3}
```

問 2

```
max=0
for i in 5 9 1 2 10 4 6 7
do
    if [ ${i} -gt ${max} ]; then
        max=${i}
    fi
done
echo ${max}
```

問 3

```
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>
#include <string.h>

int main(int argc, char* argv[])
{
    char s[256];
    printf("コマンドライン第 1 引数: %s\n", argv[1]);
    int file1 = open(argv[1], O_RDONLY);
    read(file1, s, sizeof(s));
    write(1, s, strlen(s));
    close(file1);

    return (0);
}
```

問 4

```
# (1)
touch sort.c
git add -A
git commit -a -m "ソーティング追加"
git push origin master

# (2)
git branch quicksort
git checkout quicksort
```

問 5

```
# (1)
ps u > psresult.txt

# (2)
echo "著作物"

# (3)
cat cat.txt | grep "である。"
```