

レポート提出票

科目名: 情報工学実験1

実験課題名: 課題1 テキスト処理

実施日: 2020年 7月 20日

学籍番号: 4619055

氏名: 辰川力駆

共同実験者:

_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

1 実験内容

Unix 系 OS のコマンドラインインターフェース上でテキストデータを加工する方法を学ぶ。

1.1 レポート課題 1

LaTeX コード (sample.tex) を html に変換するシェルスクリプトを作成せよ。ファイル名は “tex2html_4619055.sh” とする。以下に本課題の注意事項を示す。

- 「情報工学実験 1」という文字列は、<h2>...</h2> で囲むこと。
- itemize 環境は、html の 環境に変換すること。
- enumerate 環境、html の 環境に変換すること。
- LaTeX のコメント文は、html のコメント文に変換すること。削除してはならない。
- 作成する html の文字コードは utf-8 とすること。

1.2 レポート課題 2

課題 1 で作成した html ファイル中のコメント文 <!-- End of ol --> の位置に新たにグループを追加するシェルスクリプトを作成せよ。ファイル名は “update_html_4619055.sh” とする。シェルスクリプトの実行時の引数を用いて、グループ名や学籍番号の範囲を指定せよ。具体的には、第 1 引数にグループ名を、第 2 引数と第 3 引数に学籍番号の範囲を記述する。

1.3 レポート課題 3

課題 2 で html ファイルを更新する前に、バックアップを取るように “update_html_4619055.sh” を変更しなさい。変更後のシェルスクリプトのファイル名は、 “update_html_ver2_4619055.sh” とすること。また、“sample.html” のバックアップファイル名は “sample.html. 更新日時” とする。ここで、“update_html_4619055.sh” が実行された時刻である。

1.4 レポート課題 4

第 1 引数に “undo” という文字列を指定した場合に、更新時刻が最も新しいバックアップファイルで “sample.html” で上書きされ、元の状態を復元できるように “update_html_ver2_4619055.sh” を変更しなさい。変更後のファイル名は、“update_html_ver3_4619055.sh” とする。

2 実験結果

2.1 レポート課題1

課題1の結果を下記に示した。正常に動いている。

```
$ bash tex2html.4619055.sh
$ cat sample.html
<!DOCTYPE html>
<meta charset="utf-8">

<body>

<h2> 情報工学実験1</h2>

TeX の基本操作と簡単なレポート作成の実習を行う。

<ul style="list-style-type: circle">
  <li> 日時: 平成xx年xx月xx日 xx:xx — xx:xx</li>
  <li> 場所: 東京理科大学葛飾キャンパス講義棟xxx教室</li>
</ul>

なお、グループ編成は<u>以下の通りとする</u>。

<ol type="a">
  <li> G1 (学籍番号001 — 020)</li>
  <li> G2 (学籍番号021 — 040)</li>
  <li> G3 (学籍番号041 — 060)</li>
  <li> G4 (学籍番号061 — 080)</li>
  <!-- End of ol -->
</ol>

</body>
```

2.2 レポート課題2

課題2の結果を下記に示した。エラーハンドリングも含めて正常に動いている。

```
$ cat sample.html | tail -10

<ol type="a">
  <li> G1 (学籍番号001 — 020)</li>
  <li> G2 (学籍番号021 — 040)</li>
  <li> G3 (学籍番号041 — 060)</li>
  <li> G4 (学籍番号061 — 080)</li>
  <!-- End of ol -->
</ol>

</body>
$ bash update.html.4619055.sh
エラー: グループ名を第1引数に, 学籍番号の開始番号を第2引数に,
終了番号を第3引数に指定してください。
$ bash update.html.4619055.sh G5
エラー: 引数が足りません。
$ bash update.html.4619055.sh G5 G5 G5 G5
エラー: 引数が多すぎます。
```

```
$ bash update_html_4619055.sh G5 81 100
$ bash update_html_4619055.sh G6 101 120
$ cat sample.html | tail -12
```

```
<ol type="a">
  <li> G1 (学籍番号001 — 020)</li>
  <li> G2 (学籍番号021 — 040)</li>
  <li> G3 (学籍番号041 — 060)</li>
  <li> G4 (学籍番号061 — 080)</li>
  <li> G5 (学籍番号081 — 100)</li>
  <li> G6 (学籍番号101 — 120)</li>
<!-- End of ol -->
</ol>

</body>
```

2.3 レポート課題3

課題3の結果を下記に示した。バックアップが取れている。

```
$ ls
sample.html sample.tex tex2html_4619055.sh update_html_4619055.sh
update_html_ver2_4619055.sh
$ bash update_html_ver2_4619055.sh G7 121 140
$ ls
sample.html sample.html.20200723060707 sample.tex tex2html_4619055.sh
update_html_4619055.sh update_html_ver2_4619055.sh
```

2.4 レポート課題4

課題4の結果を下記に示した。第1引数にundoを与えると最新のバックアップに戻っている。

```
$ cat sample.html | tail -10

<ol type="a">
  <li> G1 (学籍番号001 — 020)</li>
  <li> G2 (学籍番号021 — 040)</li>
  <li> G3 (学籍番号041 — 060)</li>
  <li> G4 (学籍番号061 — 080)</li>
<!-- End of ol -->
</ol>

</body>
$ bash update_html_ver3_4619055.sh G5 81 100
$ bash update_html_ver3_4619055.sh G6 101 120
$ bash update_html_ver3_4619055.sh G7 121 140
$ ls
sample.html sample.html.20200723061544 sample.html.20200723061551
sample.html.20200723062320 sample.tex tex2html_4619055.sh
update_html_4619055.sh update_html_ver2_4619055.sh
update_html_ver3_4619055.sh
$ bash update_html_ver3_4619055.sh undo
Undo!
$ diff -u sample.html sample.html.20200723062320
$ diff -u sample.html sample.html.20200723061551
```

```

— sample.html 2020-07-23 06:24:16.706558000 +0900
+++ sample.html.20200723061551 2020-07-23 06:15:51.742743000 +0900
@@ -20,8 +20,7 @@
    <li> G3 (学籍番号041 — 060)</li>
    <li> G4 (学籍番号061 — 080)</li>
    <li> G5 (学籍番号081 — 100)</li>
-   <li> G6 (学籍番号101 — 120)</li>
- <!-- End of ol -->
+ <!-- End of ol -->
  </ol>

</body>

```

3 考察

3.1 レポート課題1

tex2html_4619055.sh を作る上で用いたものは、主に sed コマンドである。一つ一つに対し sed コマンドで書き換えて実行しようと考えた。また、最後には無駄な行を省くためにいらぬ空白行を削除して、最後に sample.html に出力した。

3.2 レポート課題2

update_html_4619055.sh を作る上で考えたのは、if のエラー分岐である。\$# で引数の個数があるので、それが3以外るとき、エラー出力するように考えた。

また、出力するときは、sed を用いて、もともと <!-- End of ol --> があった場所を、指定された文字を出力して改行した上で <!-- End of ol --> を出力することで実装している。

3.3 レポート課題3

update_html_ver2_4619055.sh は、update_html_4619055.sh を改良したものであるが、追加したのは一行だけである。具体的には、実行される前の段階で

```
cp sample.html sample.html.`date +%Y%m%d%H%M%S`
```

を追加することで、実装している。date コマンドで現在の日付や時刻を取得してそれをコピー先の名前に追加した。

3.4 レポート課題4

update_html_ver3_4619055.sh は、update_html_ver2_4619055.sh を改良したものである。第1引数に undo という文字が入っていたならば、戻す作業を行うように if 文を加えた。その後は、ls コマンドで “sample.html. 日付” となっているものだけ取得し、sort コマンドの -r オプションで降順に並べ替え、最後に head コマンドで一行だけ取得した。

4 結論

本実験を通して、シェルの基本機能及び Unix 系 OS の基本的なコマンドについて学ぶことができた。また、コマンドラインインターフェース上で sed と awk を用いたテキストデータの処理技法についても学ぶことができた。