CSSC 環境構築等についての引き継ぎ資料

2022/08/19
粕谷 俊介(s1300071@u-aizu.ac.jp)

* 特記の無い限り、以下の作業はマスターアカウントで行っている。
* シェルスクリプトの実行はlinsv等で実施すること。演習室のターミナル直で実行すると入出力周りでエラーが発生する。

事前の環境構築

概要

今年度より、Cent OSが導入されている演習室3での実施に変更となった。(実際は2021年度より変更となったが、中止となった。)そのため、2019年に使用されていたテキストと比較して、Web StormからVScodeへの変更、またmac OSからCent OSへの変更がある。事前の環境構築では、作業ディレクトリの設定や、VSCode関連の設定を行った。

情報センターとの事前のやりとりより、参加者アカウントのパスワードは紙でしか提供されないことが判明した。我々で環境構築を行うためには、参加者のアカウントにログインする必要があるが、一つ一つ人力で作業していくのは現実的ではなく、突発的に参加者全員に対してファイル配布等をしたい場合がある。そのため、シェルスクリプトでSSHを自動化して環境構築することとした。しかし、参加者全員のパスワードを書き起こすのは煩雑で、全員分のパスワードを一括して保管することはセキュリティ面で懸念がある。そのため、マスターアカウントから参加者アカウントに公開鍵認証を設定した。

事前の環境構築で作業するシェルスクリプトでは、サンプルディレクトリ(Sample/)のコピーと、VSCode設定ファイルのコピー(VSCode/settings.json)、最新のNode.js v16.16.0のプログラムのコピーおよびその設定が含まれる。

環境構築スクリプト実行のための準備

まず、参加者のアカウントにログインするための公開鍵認証の設定をする。情報センターとのやりとりより、ファイルのコピーは情報センターで実施可能とのことだったので、公開鍵の登録を依頼した。

1. 公開鍵 / 秘密鍵の生成
   以下のように、公開鍵 / 秘密鍵のペアを生成し、情報センターへの依頼用ディレクトリを作成してコピーする。

```
$ mkdir ~/.ssh
$ cd ~/.ssh
$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key
(/home/seminar/g2java/g2java00/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in id_ed25519.
Your public key has been saved in id_ed25519.pub.
```

```
The key fingerprint is:
SHA256:XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX g2java00@std3dc49
The key's randomart image is:
+--[ED25519 256]--+
|                 |
|                 |
|                 |
|                 |
|      (省略)      |
|                 |
|                 |
|                 |
|                 |
+----[SHA256]-----+
$ ls -la id_ed25519*
-rw------- 1 g2java00 guest02 464 Aug 12 14:38 id_ed25519
-rw-r--r-- 1 g2java00 guest02  99 Aug 12 14:38 id_ed25519.pub
$ mkdir ~/sshconfig
$ touch ~/sshconfig/authorized_keys
$ cat id_ed25519.pub >> ~/sshconfig/authorized_keys
```

2. 情報センターにコピーを依頼
   上記で作成した公開鍵を、参加者の所定の場所にコピーするように以下のように依頼した。

```
マスタアカウントの、/home/seminar/g2java/g2java00/sshconfig/authorized_keys
を、/home/seminar/g2java/g2java[01-25]/.ssh/ へコピーをお願い致します。なお、コピ
ー先の参加者アカウントには、.sshディレクトリはないので、作成をお願い致します。
コピー作業依頼時間は、15日の13時に間に合うようにしていただければ幸いです。上述したファイ
ル(authorized_keys)は今後変更する予定はありません。
```

参加者のアカウントに公開鍵認証でSSHは以下でできる。途中、秘密鍵のパスワードが求められる。なお、XXはユーザーアカウントの数字。

```
ssh -i ~/.ssh/id_ed25519 g2javaXX@sshgate.u-aizu.ac.jp
```

環境構築の実行

ディレクトリ構成

```
.
├── Sample (講習で使用するプログラムのサンプルが入っている)
│   ├── 5.1sample.html
│   ├── img
│   ├── text.html
│   └── text_key.html
├── VSCode (VSCodeの設定ファイルが入っている)
│   └── settings.json
```

```
├── auto_setting.sh
├── node_v16.16.0(Node.jsのプログラムが入っている。これはNode.js公式ページから最新のLTS版をダウンロードした。)
│   ├── CHANGELOG.md
│   ├── LICENSE
│   ├── README.md
│   ├── bin
│   ├── include
│   ├── lib
│   └── share
└── user_list.txt
```

まず、シェルスクリプトが入っているディレクトリと同じ階層にSampleディレクトリを準備する。ここには、講習で使用するサンプルファイルを入れる。今回は5.1のCreateJSのゲームのサンプルや、講習で作成するゲームの完成形などを入れた。

次に、VSCodeの設定ファイルを準備する。ここには、LiveServer使用時のデフォルトブラウザの設定等をした。VSCode関連の設定はここに追記すればできる。

VSCode/settings.json

```
{
"liveServer.settings.CustomBrowser": "firefox",
"files.autoSave": "afterDelay"
}
```

次に、Node.jsの最新版を準備する。テキスト第4章にて、JavaScriptの基本を学ぶ。テキストではブラウザのデベロッパーツールを使うようになっているが、デモの文字サイズの都合上、ターミナルでNode.jsのREPLを使用した。演習室のPCにはNode.jsが既に入っているが、v10.23.1と古かったため最新のLTSを用意した。Node.js公式ページからダウンロードして、tar.xzを解凍する。

(実行例)

```
cd ~/
wget https://nodejs.org/dist/v16.17.0/node-v16.17.0-linux-x64.tar.xz
tar -xvJf node-v16.17.0-linux-x64.tar.xz
mv node-v16.17.0-linux-x64 ~/auto_setting/node-v16.17.0
```

最後に、user_list.txtを準備する。これはスクリプトを実行する対象の参加者アカウントのユーザー名のリストである。

user_list.txt (例)

```
g2java01
g2java02
g2java03
g2java04
```

```
g2java05
g2java06
g2java07
g2java08
g2java09
g2java10
g2java11
g2java12
g2java13
g2java14
g2java15
g2java16
g2java17
g2java18
g2java19
g2java20
g2java21
g2java22
g2java23
g2java24
g2java25
```

シェルスクリプトを実行する。なお、演習室の端末で実行するとパスワード入力の部分で標準入出力のエラーが発生するので、linsv等でSSH接続して実行する。また、デフォルトではユーザーにも実行権限が付与されていないため、実行権限を付与する。

(実行例)

```
$ ssh linsv
g2java00@linsv's password:
(略)
$ cd ~/auto_setting/
$ chmod 755 auto_setting.sh
$ ./auto_setting.sh
```

auto_setting.sh

```
#!/usr/bin/env bash

# Auto Setting File for Computer Science Summer Camp 2022
#
# CSSC用の自動設定シェルスクリプトです。このシェルスクリプト単体では動作しません。
# Created by KASUYA Shunsuke (s1300071@u-aizu.ac.jp)



timeout_nomal=5
timeout_long=15
# 秘密鍵のパスワードを指定
```

```
PASSWORD=SECRET_KEY_PASSWORD

cat user_list.txt | while read username
do
 expect -c "
    set timeout ${timeout_long}
    # SCPでサンプルをコピー
    spawn scp -i ~/.ssh/id_ed25519 -r -o StrictHostKeyChecking=no -C
./Sample ${username}@sshgate.u-aizu.ac.jp:~/
    expect -re \"Enter passphrase for key .*:\"
    send \"${PASSWORD}\n\"
    # SCPでVSCodeのsetings.jsonをコピー
    spawn scp -i ~/.ssh/id_ed25519 -r -o StrictHostKeyChecking=no -C
./VSCode/settings.json ${username}@sshgate.u-
aizu.ac.jp:~/.config/Code/User/settings.json
    expect -re \"Enter passphrase for key .*:\"
    send \"${PASSWORD}\n\"
    # SCPでNode.js v16.16.0をコピー
    spawn scp -i ~/.ssh/id_ed25519 -r -o StrictHostKeyChecking=no -C
./node_v16.16.0 ${username}@sshgate.u-aizu.ac.jp:~/
    expect -re \"Enter passphrase for key .*:\"
    send \"${PASSWORD}\n\"
    # 公開鍵認証方式でSSH
    spawn ssh -i ~/.ssh/id_ed25519 -o StrictHostKeyChecking=no
${username}@sshgate.u-aizu.ac.jp
    expect -re \"Enter passphrase for key .*:\"
    send \"${PASSWORD}\n\"
    expect \"$\"
    # Node v16を使用するようにalias
    send \"echo alias node='~/node_v16.16.0/bin/node' >> ~/.bashrc\n\"
    expect \"$\"
    # 一旦、VSCodeを実行させないと、~/.vscodeディレクトリが作成されないらしい
    send \"code &\n\"
    expect \"$\"
    # エスケープが気持ち悪いが、VSCodeの日本語化設定
    send \"sed -i \\\"$ i ,\\\\\\\"locale\\\\\\\": \\\\\\\"ja\\\\\\\"\\\"
~/.vscode/argv.json\n\"
    expect \"$\"
    set timeout ${timeout_long}
    # VSCodeの拡張機能を追加
    send \"code --install-extension 'MS-CEINTL.vscode-language-pack-ja' --
install-extension 'ritwickdey.LiveServer'\n\"
    expect \"$\"
    set timeout ${timeout_nomal}
    send \"cd ~/\n\"
    expect \"$\"
    send \"mkdir Project\n\"
    expect \"$\"
    send \"exit\"
    exit 0
  "
done
```

数個の適当に選んだ参加者アカウントにSSHログインし、正しくディレクトリ等が作成されているか確認した。
事前の環境構築は以上となる。

実施中の運用

概要

運用中は参加者のサポートと、グループ開発中用のシンボリックリンク作成、グループ開発ファイルの権限変更、バックアップ等を行った。

グループ開発では、グループの参加者が同一のディレクトリを参照できるように、シンボリックリンクを作成した。2019年のテキストでは、参加者のグループ開発用ディレクトリからグループのリーダーのディレクトリにコピーし、必要であればその逆を行って同期をとっていたが、参加者がいちいちコマンドを叩くのは難しく、後にファイルを回収する上でも不便だと考えた。そこで、今年度は各参加者の`TeamProject`をマスタアカウントの`~/GraduationProjects/Group0X/`にシンボリックリンク貼ることにした。グループ開発前に、参加者のユーザー名とグループを紐付けたテキストファイルを作り、シェルスクリプトでシンボリックリンクを作成した。

グループ開発中は、グループ開発で作成したファイルの権限変更をシェルスクリプトで行った。シンボリックリンク作成時に、卒業制作が入るディレクトリは`umask 002`を設定したが、VSCodeの新規ファイル作成機能では適用されないことがわかった。そのため、強引ではあるがグループ開発実施中は各参加者のアカウントにSSHし、`TeamProject(実態は/home/seminar/g2java/g2java00/GraduationProjects/Group0X/)`の中身を`chmod 775 *`するようなシェルスクリプトを実行していた。

また、グループ開発の途中から5分毎のバックアップを取るようにした。このグループ開発の方法では、Gitと違いバージョン管理ができなく、削除してしまったファイルは復帰できない。誤ってファイルを削除してしまっても対応できるように、5分毎にバックアップを取るシェルスクリプトを回した。

グループ開発用シンボリックリンクの作成の準備

まず、マスタアカウントにグループ開発用ディレクトリを作成する。これは事前の環境構築でやってしまってもよい。

ディレクトリ構成例

```
~/
── GraduationProjects
    ├── Group01
    ├── Group02
    ├── Group03
    └── Group04
```

コマンド実行例

```
cd ~
mkdir GraduationProjects
```

```
chmod 775 GraduationProjects
cd GraduationProjects
mkdir Group{01..04}
chmod 775 Group*
```

グループ開発用シンボリックリンクの作成

参加者のユーザー名とグループを対応付けるテキストファイルを作成する。その後、シェルスクリプトを実行して、シンボリックリンクを作成する。

1. 参加者のユーザー名とグループを対応付けるテキストファイルを以下の通りに作成する。グループの切り替わりは空行で表している。シェルスクリプトと同じ階層のディレクトリに作成する。

テキストファイル作成例

```
g2java05
g2java07
g2java01
g2java03
g2java04

g2java10
g2java08
g2java06
g2java02

g2java16
g2java12
g2java21
g2java20
g2java09

g2java13
g2java14
g2java19
g2java17
g2java18
```

2. シェルスクリプトを実行する。なお、シェルスクリプトに実行権限が付与されていることを確認する。

group_setting.sh

```
#!/usr/bin/env bash

# Group Setting File for Computer Science Summer Camp 2022
#
# CSSC用のグループ開発用シェルスクリプトです。
# Created by KASUYA Shunsuke (s1300071@u-aizu.ac.jp)
```

```
timeout_nomal=5
timeout_long=15
# 秘密鍵のパスワードを指定
PASSWORD=SECRET_KEY_PASSWORD
groupNo=1

rm group_setting.result
echo -e "Group:${groupNo}" >> group_setting.result

cat group_list.txt | while read username
do
 echo ${groupNo}
 # 読み込んだ行が空行だったら、次のグループへ
 if [ -z "$username" ]; then
        $((groupNo++))
        echo -e "\nGroup:${groupNo}" >> group_setting.result
        continue
 fi
 expect -c "
    set timeout $timeout_nomal
    spawn ssh -i ~/.ssh/id_ed25519 -o StrictHostKeyChecking=no
${username}@sshgate.u-aizu.ac.jp
    expect -re \"Enter passphrase for key .*:\"
    send \"${PASSWORD}\n\"
    expect \"$\"
    # 参加者のTeamProjectとマスタアカウントのグループのディレクトリをシンボリックリン
ク
    send  \"ln -s
/home/seminar/g2java/g2java00/GraduationProjects/Group0${groupNo}
~/TeamProject\n\"
    expect \"$\"
    send \"exit\"
    exit 0
  "
 echo -e "${username}" >> group_setting.result
done
```

3. 任意の参加者のアカウントにSSHログインして、シンボリックリンクが作成されていることを確認する。

コマンド実行例

```
$ ssh g2java05@sshgate.u-aizu.ac.jp -i ~/.ssh/id_ed25519
$ ls -la TeamProject
lrwxrwxrwx  1 g2java05 guest02   56 Aug 17 13:15 TeamProject ->
/home/seminar/g2java/g2java00/GraduationProjects/Group01
```

赤字黒背景で点滅していたりしていると、シンボリックリンク先へアクセスできないことを示している。この場合は、リンク先の権限などを確認すること。

グループ開発中の権限変更シェルスクリプト

上述したとおり、参加者がVSCodeで作成したファイルの権限に、グループの書き込み権限がない。また`umask`コマンドで設定したが適用されなかった。そのため、グループ開発実行中は常に以下のシェルスクリプトを実行していた。この点は今後検討するべき課題である。

group_chmod.sh

```
#!/usr/bin/env bash

# Group Project Chmod Script File for Computer Science Summer Camp 2022
#
# CSSC用のグループ開発の権限変更用シェルスクリプトです。
# Created by KASUYA Shunsuke (s1300071@u-aizu.ac.jp)

timeout_nomal=5
timeout_long=15
# 秘密鍵のパスワードを指定
PASSWORD=SECRET_KEY_PASSWORD

while true
do
  cat user_list.txt | while read username
  do
  expect -c "
      set timeout ${timeout_nomal}
      spawn ssh -i ~/.ssh/id_ed25519 -o StrictHostKeyChecking=no
${username}@sshgate.u-aizu.ac.jp chmod -Rf 775 ./TeamProject/*
      expect -re \"Enter passphrase for key .*:\"
      send \"${PASSWORD}\n\"
      expect \"$\"
      exit 0
    "
  sleep 1
  done
  sleep 5
done
```

グループ開発中のバックアップシェルスクリプト

上述したとおり、グループ開発の途中からバックアップを取り始めた。

1. バックアップ先ディレクトリの作成

コマンド例(マスタアカウントで実行)

```
cd ~/GraduationProjects/
mkdir snapshot
```

2. バックアップ対象グループ名リストの作成
   シェルスクリプトがある場所と同じ階層に`group_snapshot_list.txt`を作成する。

group_snapshot_list.txt

```
Group01
Group02
Group03
Group04
```

3. バックアップシェルスクリプトの実行
   このスクリプトを実行することで、マスタアカウントの`~/GraduationProjects/snapshot/`ディレクトリ以下に、`dd_hh:mm`の形式でディレクトリが作成され、その下に`Group0X.tar.gz`でその時刻のバックアップが保存される。

backupGp.sh

```
#!/usr/bin/env bash

# Group Project Backup Script File for Computer Science Summer Camp 2022
#
# CSSC用のグループ開発時バックアップ用シェルスクリプトです。
# Created by KASUYA Shunsuke (s1300071@u-aizu.ac.jp)

while true
do
  DATE=$(date '+%d_%R')
  mkdir ~/GraduationProjects/snapshot/${DATE}
  cat group_snapshot_list.txt | while read groupname
do
      cd  ~/GraduationProjects/snapshot/${DATE}/
      # tar.gzで圧縮してバックアップ
      tar -czvf ${groupname}.tar.gz ../../${groupname}/
      chmod 444 ${groupname}.tar.gz
done
# 300秒(5分)待機
sleep 300
done
```

本来であればcronで5分毎にやるのが良い方法だが、演習室のPCにcronが設置できなかったため、この方法を採用している。

全課程終了時の事後バックアップ・作業

概要

CSSCの修了式が終わったら、事後バックアップと、HP担当者への卒業生作品の引き渡しがある。2019年度までは事後バックアップの一部を情報センターに依頼していたが、今年度から事後バックアップはインスト

ラクターがすべて実施することになった。作業が非常に煩雑なため、次年度以降は情報センターになんとか依頼することを強く推奨する。
まず、バックアップデータの仮保存先として、検証用のアカウントを使用する。各参加者のアカウントから直にバックアップ先へアクセスすることはできない。そのため、検証用アカウントに777なパーミッションのディレクトリを用意し、そこへバックアップする。その後、学籍番号のアカウントにてそのディレクトリにアクセスし、cpコマンドでコピーすることによって所有者を学籍番号のアカウントに変更する。そして、chgrpコマンドで`css-camp`グループに変更する。
また、例年のバックアップはそのまま保存しているが、情報センターよりバックアップ先である`/home/projects/css-camp`以下のサイズが大きいと指摘された。そのため、今年度は`tar.gz`形式で圧縮して保存した。

事後バックアップ作業の準備

1. グループ追加の依頼
   事後バックアップの一部の作業にて、`css-camp`グループに所属していないとできない作業がある。そこで、情報センターに学籍番号のアカウントについて`css-camp`グループへの所属を依頼する。なお、情報センター曰く、`css-camp`グループへの参加はゲストアカウントではできないため、学籍番号のアカウントをグループに追加するように依頼する。

2. 検証用アカウントへディレクトリの作成
   検証先アカウントへバックアップデータの仮保存先のディレクトリを作成する。検証用アカウントの例として、`g1cssc3`を使用する。なお、グループが違うため、パーミッションは777に設定している。セキュリティ的に懸念があるため、バックアップ作業終了後はパーミッションを変更することを推奨する。

コマンド例(`g1cssc3`のアカウントで実行)

```
mkdir ~/backup
chmod 777 ~/backup
```

3. バックアップ先ディレクトリの作成
   バックアップ先(`/home/project/css-camp/`)へディレクトリを作成する。各年度のディレクトリを作成し、その下に`student`ディレクトリと`gp`ディレクトリを作成する。`student`ディレクトリにはバックアップファイルが入る。`gp`ディレクトリには卒業制作のバックアップが入る。なお、`student`ディレクトリと`gp`ディレクトリの下にコースごとにディレクトリを用意する。これらのディレクトリには再帰的にパーミッションを775に設定する。また、グループを`css-camp`に変更する。

コマンド例(`css-camp`グループに所属している学籍番号のアカウントで実行)

```
$ cd /home/project/css-camp/
$ mkdir 20XX
$ chmod 775 20XX
$ chgrp css-camp 20XX
$ cd 20XX
$ mkdir {student,gp}
$ chmod 775 student
$ chmod 775 gp
```

```
$ chgrp css-camp student
$ chgrp css-camp gp
$ mkdir ./student/g2java
$ mkdir ./gp/g2java
$ chmod 775 ./student/g2java
$ chmod 775 ./gp/g2java
$ chgrp css-camp ./student/g2java
$ chgrp css-camp ./gp/g2java
```

事後バックアップ

全てが終了したところで事後バックアップ作業を開始する。

1. シェルスクリプトの実行
   シェルスクリプトと同じ階層に、バックアップ対象のアカウントを指定したテキストファイルを用意する。(これは環境構築スクリプトで行ったファイルと同一である。)

backup.sh

```
#!/usr/bin/env bash

# Backup Script for Computer Science Summer Camp 2022
#
# CSSCバックアップ用スクリプト. 777なパーミッションの検証用アカウントのbackupディレクトリに、tar.gzで保存される。
# この後、css-campグループが付与されているアカウントから/home/project/20XX/students/g2java/にcpすること。
# Created by KASUYA Shunsuke (s1300071@u-aizu.ac.jp)

timeout_long=10
timeout_tar=30
timeout_nomal=5

# 秘密鍵のパスワードを指定
PASSWORD=SECRET_KEY_PASSWORD


cat user_list.txt | while read username
do
 expect -c "
    set timeout ${timeout_long}
    spawn ssh -i ~/.ssh/id_ed25519 -o StrictHostKeyChecking=no ${username}@sshgate.u-aizu.ac.jp
    expect -re \"Enter passphrase for key .*:\"
    send \"${PASSWORD}\n\"
    expect \"$\"
    set timeout ${timeout_tar}
    send \"tar -cvzf /home/seminar/g1cssc/g1cssc3/backup/${username}.tar.gz ../${username}/\n\"
    expect \"$\"
    send \"exit\"
```

```
    exit 0
  "
done
```

2. バックアップデータ仮保存先から保存先へコピー
   css-campグループが付与されている、学籍番号のアカウントでログインして、準備で作成したバックアップデータ仮保存先にバックアップデータが保存されていることを確認する。実際にいくつか解凍して確かめてみるのも良い。その後、バックアップ先へコピーする。

コマンド例(s1XXXXXXのアカウントでログイン)

```
$ cd /home/seminar/g1cssc/g1cssc3/backup
$ ls -la
total 2492196
drwxr-xr-x  4 g1cssc3  guest01      4096 Aug 19 12:16 .
drwxrwxrwx 33 g1cssc3  guest01      8192 Aug 19 12:21 ..
-rw-r--r--  1 g2java01 guest02  82051072 Aug 19 11:40 g2java01.tar.gz
-rw-r--r--  1 g2java02 guest02 149946368 Aug 19 11:41 g2java02.tar.gz
-rw-r--r--  1 g2java03 guest02  70778880 Aug 19 11:42 g2java03.tar.gz
-rw-r--r--  1 g2java04 guest02  81526784 Aug 19 11:43 g2java04.tar.gz
-rw-r--r--  1 g2java05 guest02 207618048 Aug 19 11:43 g2java05.tar.gz
(省略)
$ cp * /home/project/css-camp/2022/students/g2java/
```

3. コピーしたバックアップデータのグループを変更する
   コピーすることでファイルの所有者は学籍番号のアカウントになるが、グループはstudentのままである。特にこのままでも問題はないが、css-campのグループがあるのであればそのグループにしたいと思うので、変更する。また、ファイル追記を禁止するために、パーミッションを444へ変更する。

コマンド例(css-campグループに所属している学籍番号のアカウントでログイン)

```
cd /home/project/css-camp/2022/students/g2java
chmod 444 *
chgrp css-camp *
```

卒業制作のバックアップ

上では参加者のアカウント丸々をバックアップしたが、卒業制作のバックアップを別ディレクトリにする必要がある。

1. 卒業制作をバックアップする
   卒業制作の製作中に回していたバックアップのシェルスクリプトをもう一度実行する。一通りバックアップをしたらCtrl-Cで止めてしまって良い。

2. 卒業制作のバックアップをバックアップデータ仮保存先にコピーする
   権限の関係で、こちらも直接バックアップ先に保存することはできない。上と同じように仮保存先にコピーする。

コマンド例

```
$ cd ~/GraduationProjects/snapshot
$ ls -la
ls -la
total 316
drwxr-xr-x 78 g2java00 guest02 8192 Aug 19 10:40 .
drwxrwxr-x  8 g2java00 guest02 4096 Aug 19 10:27 ..
(省略)
drwxr-xr-x  2 g2java00 guest02 4096 Aug 19 08:55 19_08:55
drwxr-xr-x  2 g2java00 guest02 4096 Aug 19 10:00 19_10:00
drwxr-xr-x  2 g2java00 guest02 4096 Aug 19 10:40 19_10:40
$ cd cd 19_10\:40/  (←最新のバックアップデータ)
$ ls -la
total 29120
drwxr-xr-x  2 g2java00 guest02     4096 Aug 19 10:40 .
drwxr-xr-x 78 g2java00 guest02     8192 Aug 19 10:40 ..
-r--r--r--  1 g2java00 guest02  6448479 Aug 19 10:40 Group01.tar.gz
-r--r--r--  1 g2java00 guest02  1696218 Aug 19 10:40 Group02.tar.gz
-r--r--r--  1 g2java00 guest02 14308778 Aug 19 10:40 Group03.tar.gz
-r--r--r--  1 g2java00 guest02  7205516 Aug 19 10:40 Group04.tar.gz
$ cp * /home/seminar/g1cssc/g1cssc3/backup
```

3. バックアップデータ仮保存先から保存先へコピー
   上と同様に作業する。

コマンド例(`css-camp`グループに所属している学籍番号のアカウントで実行)

```
$ cd /home/seminar/g1cssc/g1cssc3/backup
$ ls -la Group0*
drwxr-xr-x 3 g1cssc3  guest01     4096 Aug 19 12:17 .
drwxr-xr-x 4 g1cssc3  guest01     4096 Aug 19 12:16 ..
-r--r--r-- 1 g2java00 guest02  6448479 Aug 19 12:17 Group01.tar.gz
-r--r--r-- 1 g2java00 guest02  1696218 Aug 19 12:17 Group02.tar.gz
-r--r--r-- 1 g2java00 guest02 14308778 Aug 19 12:17 Group03.tar.gz
-r--r--r-- 1 g2java00 guest02  7205516 Aug 19 12:17 Group04.tar.gz
$ cp * /home/project/css-camp/2022/gp/g2java/
```

4. コピーしたバックアップデータのグループを変更する
   コピーすることでファイルの所有者は学籍番号のアカウントになるが、グループは`student`のままである。特にこのままでも問題はないが、`css-camp`のグループがあるのであればそのグループにしたいと思うので、変更する。また、ファイル追記を禁止するために、パーミッションを444へ変更する。

コマンド例(`css-camp`グループに所属している学籍番号のアカウントで実行)

```
cd /home/project/css-camp/2022/gp/g2java
chmod 444 *
chgrp css-camp *
```

HP担当者様へのデータの引き渡し

今年度はHP担当者様へデータの引き渡しのみを行った。卒業制作のデータはマスターアカウントに集約されているので、卒業制作のデータをZIPで圧縮してメールで送信した。

その他引き継ぎ事項

実施中について

受講生の初回ログイン

参加者が初回ログインするときに、一般的なPCと同様に電源ボタンを押してシャットダウンする事案が2件発生したため、ログインする前にその旨を注意することが必要であった。なお、そのうち1つのPCにてキーボードが認識しない問題が発生し、SSBが対応する事となった。

「第4章 JavaScriptの基本」の実行方法について

JavaScriptの基本としてNode.jsのREPLを用いて講習する場面があるが、戻り値をいちいち表示するため戸惑う場面が一部あった。また、打ち間違えたときにブロックの先頭から入力し直さないといけなくなる。そのため、外部ファイルとして`.js`ファイルを作り、それをNode.jsで実行させるのがいいのではないかと考える。

プログラムの無限ループに伴うフリーズについて

演習中に、受講生の一部が無限ループを発生してしまい、PCがフリーズする事案が2件あった。マウス・キーボードの操作も受け付けない状態だったため、SSHのリモートログインができずkillコマンドも発行できなかったため、やむを得ず予備アカウントを使用した。同じアカウントで別の端末にログインしようとすると二重ログインとなり、Firefoxが起動しないため、予備アカウントを使用した。なお、その際はlinsvにSSHをしてファイルの移行を行った。

講習中に演習室にリモートログインしているユーザーがいたため、一部のPCの負荷が高くなった問題

講習中にSSHでリモートログインをしている教員がいて、一部のPCで負荷が高くなり重くなる事案があった。情報センターに、「std3, std4を講習で利用するのでリモートログインは避けてほしい」旨を周知していただいた。

テキストにそこそこミスがある問題

講義テキストにそこそこミスがある。そのあたりの引き継ぎは別の人がやってくれている(はずである)が、訂正していただきたい。

その他

今年度はかなりシェルスクリプトを多用して、自動化をした。
この引き継ぎ資料で不明な点がある場合は、何なりとご連絡ください。
また、スクリプトもZIPファイルで同梱しているはずだが、なぜか紙の引き継ぎ資料しかなく、またはこのPDFだけしかなく、電子データとしてのスクリプトファイルがなくて困っている等の場合でもご連絡くださ

い。たぶんもっています。
s1300071@u-aizu.ac.jp
粕谷 俊介