

全国高校生プログラミングコンテスト

CHaserOnline

ステップアップヒント3

12 クライアント起動用シェルスクリプトの作成

クライアントを起動するには大変多くの引数やオプションが必要です。これらを毎回入力するのは大変なことですし、また間違いのもとにもなります。

それらを解消するシェルスクリプトを作成することで簡単にクライアントを起動できるようになります。テキストエディタを利用し次のように入力してください。

なお、ユーザ名・パスワード・プロキシサーバアドレスおよびプロキシサーバのポート番号は各自のものに差し替えて記入してください。

例1) 毎回ルーム番号を入力する例(Client01.sh)

```
#!/bin/sh
#./CHaserOnlineClient2017public003-2.o
    http://localhost:8080/CHaserOnline003/user/ -u cool -p cool

#./CHaserOnlineClient2017public003-2.o
    http://www7019ug.sakura.ne.jp:80/CHaserOnline003/user/ -u cool -p cool

./CHaserOnlineClient2017public003-2.o
    http://www7019ug.sakura.ne.jp:80/CHaserOnline003/user/
    -u cool -p cool -x 192.168. 〇〇〇.〇〇〇:8080
```

1行ずつ説明します。

```
#!/bin/sh
```

1行目はこれからシェルスクリプトが始まることを示します。

```
#./CHaserOnlineClient2017public003-2.o
    http://localhost:8080/CHaserOnline003/user/ -u cool -p cool
```

改行せずに空白
をあける

改行する

次の行は左はじに「#」がついています。「#」をつけるとコメント行にすることができます。

コメントを外した場合は、サーバを起動しているのと同じパソコンでクライアントを動かす場合のコマンドになっています。

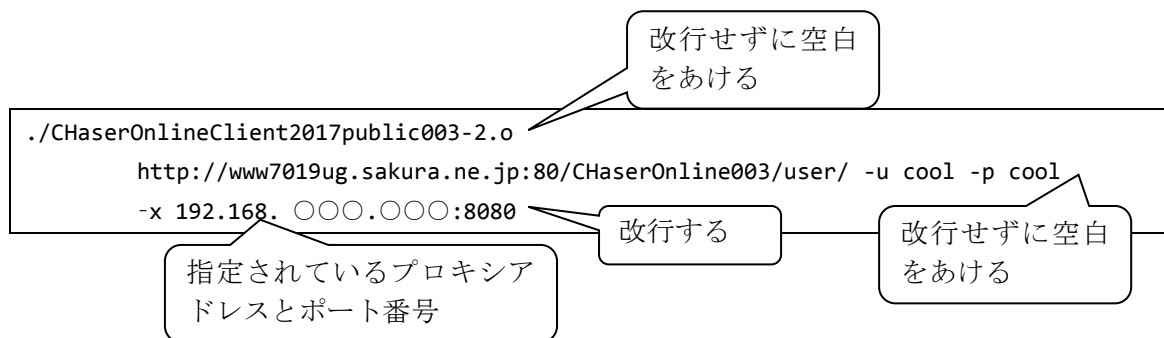
なお、紙面の都合上改行していますが、実際には改行せずに1行で記入してください。

```
#./CHaserOnlineClient2017public003-2.o
    http://www7019ug.sakura.ne.jp:80/CHaserOnline003/user/ -u cool -p cool
```

改行せずに空白
をあける

改行する

次の行もコメントになっていますが、自宅などプロキシサーバを経由せずに接続するコマンドです。



次の行は「#」がついていません。実際に動作させるのはこの行だけとなります。学校などプロキシサーバ経由でインターネットへ接続している場合のコマンドです。

クライアントを実行する環境に合わせて「#」を付け替えることで、簡単に接続先を変更することができます。

例2) ルーム番号を順番に接続する例(Client02.sh)
※ルームがあいているかどうかの確認が別途必要です。

```
#!/bin/sh
for room in 1018 1019 1020
do
    #./CHaserOnlineClient2017public003-2.o
        http://localhost:8080/CHaserOnline003/user/ -u cool -p cool -r $room

    #./CHaserOnlineClient2017public003-2.o
        http://www7019ug.sakura.ne.jp:80/CHaserOnline003/user/ -u cool -p cool -r
$room

    ./CHaserOnlineClient2017public003-2.o
        http://www7019ug.sakura.ne.jp:80/CHaserOnline003/user/
        -u cool -p cool -x 192.168.30.000:8080 -r $room

    sleep 3
done
```

このシェルスクリプトは「接続がうまくいく」ことを前提に書かれています。
それでは1行ずつ説明します。

```
#!/bin/sh
```

1行目はこれからシェルスクリプトが始まることを示します。

```
for room in 1018 1019 1020
do
    .
    .
    .
done
```

2行目、3行目および10行目はセットで使用します。

2行目の「for」コマンドで変数「room」の中に「in」以降に列記した文字を順に代入します。変数「room」へ文字を代入するたびに、3行目から10行目の「do~done」を繰り返します。

```
#!/CHaserOnlineClient2017public003-2.o
    http://localhost:8080/CHaserOnline003/user/ -u cool -p cool -r $room
```

4行目はローカル接続用に準備されたものになります。「-r \$room」とすることで「for コマンド」で順次変化する変数「room」をオプションに指定しています。

```

#./CHaserOnlineClient2017public003-2.o
    http:// www7019ug.sakura.ne.jp:80/CHaserOnline003/user/
    -u cool -p cool -r $room

./CHaserOnlineClient2017public003-2.o
    http:// www7019ug.sakura.ne.jp:80/ CHaserOnline003/user/
    -u cool -p cool -x 192.168. 〇〇〇.〇〇〇:8080 -r $room

```

改行せずに空白をあける

改行する

改行せずに空白をあける

この行はルーム番号を指定している部分以外は例1と同様になります。

```
sleep 3
```

この行は次のコマンドまでの待ち時間を設定します。数字は秒数を表すので3秒待ってループに戻ります。

Client02.sh はクライアントがある程度出来上がったときに、複数のルームで動作検証をするときに実行すると良いでしょう。

次にシェルスクリプトを実行可能な属性に変更します。

端末のコマンドラインにて次のコマンドを実行します。

```

chmod 700 Client01.sh
および
chmod 700 Client02.sh

```

シェルスクリプトを実行するには端末のコマンドラインで以下のように実行します。

```

./Client01.sh
または
./Client02.sh

```

Client01.sh や Client02.sh などファイル名の前に「./」を入れるのを忘れないようにしてください。

なお、作品を第二次予戦に提出するときには、シェルスクリプトからの実行は必須になります。(Windows 環境ではバッチファイルで代用してください)

13 サンプルプログラム 3

サンプルプログラム 3 は、次のようになっています。

(1) GetReadyMove を発行するプログラム

```
while(1){
    /*-----
    GetReadyMove を発行する
    -----*/
    do{
        printf("%n%n%ndeb191 GetReady%n");

        strcpy(param, "command1=");
        if(ActionCount == 9){
            if(mode==51){
                if(ActionReturnNumber[6]==5){
                    GetReadyMode = 1;
                }
            }
            else if(mode==53){
                if(ActionReturnNumber[6]==5){
                    GetReadyMode = 1;
                }
                else if(ActionReturnNumber[3]==1){
                    GetReadyMode = 3;
                }
                else if(ActionReturnNumber[3]==2){
                    GetReadyMode = 1;
                }
            }
        }
    }
    else{
        GetReadyMode = 0;
    }
}
```

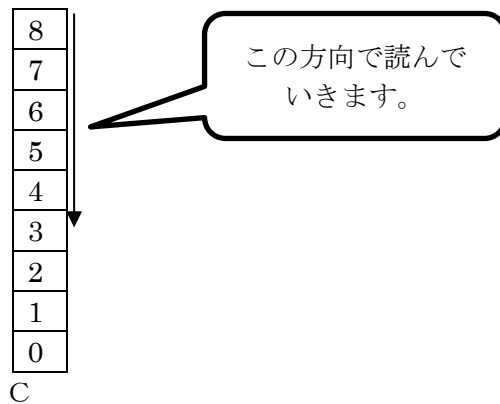
- ① Action の周囲情報が 9 個の時に判定する。

```
・
・
if(ActionCount == 9)
```

② pu3su の周囲情報の 6 つ目が化石なら上へ

```
•  
•  
if(mode==51){  
    if(ActionReturnNumber[6]==5){  
        GetReadyMode = 1;  
    }  
•  
•
```

「ActionReturnNumber[6]」は矢印の方向から考えて行くので「2 番目」の周囲情報をつかむことができる。



③ 下記の例にて「p 1 3 s 1」の周囲情報を読んで、動作します。

```
•  
•  
  
    else if(mode==53){  
        if(ActionReturnNumber[6]==5){  
            GetReadyMode = 1;  
        }  
        else if(ActionReturnNumber[3]==1){  
            GetReadyMode = 3;  
        }  
        else if(ActionReturnNumber[3]==2){  
            GetReadyMode = 1;  
        }  
    }  
•  
•
```

動作例

0	1	2	3	4	5	6	7	8	C
---	---	---	---	---	---	---	---	---	---

6番が化石の時は上へ移動します。

3番がターゲットの時は左へ移動します。

3番が土の時は上へ移動します。

(2) 「Action を発行する」プログラム

```
/*-----  
   Action を発行する  
-----*/  
do{  
    strcpy(param, "command2=");  
    if(count == 9){  
        if(returnNumber[5]==5){  
            mode = 51;  
        }  
        else if(returnNumber[3]==3){  
            mode = 57;  
        }  
        else if(returnNumber[1]==2){  
            mode = 51;  
        }  
        else if(returnNumber[1]==1){  
            mode = 53;  
        }  
        else if(returnNumber[0]==7){  
            mode = 53;  
        }  
        else{  
            mode = 55;  
        }  
    }  
    else{  
        if(mode == 55){  
            mode = 53;  
        }  
    }  
  
    switch(mode){  
        case 1:  
            strcat(param, "wu");  
            break;
```

```
        case 5:
            strcat(param, "wr");
            break;
        case 7:
            strcat(param, "wd");
            break;

        case 51:
            strcat(param, "pu3su");
            break;

        case 53:
            strcat(param, "pl3sl");
            break;

        case 55:
            strcat(param, "pr3sr");
            break;

        case 57:
            strcat(param, "pd3sd");
            break;

        default:
            strcat(param, "wr");
    }
    .
    .
    .
```

① GetReadyMove の周囲情報が 9 個の時

```
if(count == 9){
```


② 各移動プログラム

移動プログラム	解説
if (returnNumber[5]==5) { mode = 51; }	右が化石なら pu3su (上にアイテムを置き 右に行く)
else if(returnNumber[3]==3){ mode = 57; }	左がアイテムなら pd3sd (下にアイテムを 置き左に行く)
else if(returnNumber[1]==2){ mode = 51; }	上が土ならば pu3su (上にアイテムを置き 右へ行く)、ターゲットなら pl3sl (左にア イテムを置き上へ行く)
else if(returnNumber[1]==1){ mode = 53; }	
else{ mode = 55; }	その他は pr3sr (右にアイテムを置き下へ 行く)
else{ if(mode == 55){ mode = 53; }	周囲情報が9個以外の時

③ put3&search の追加

case 51:	strcat(param, "pu3su"); break;
case 53:	strcat(param, "pl3sl"); break;
case 55:	strcat(param, "pr3sr"); break;
case 57:	strcat(param, "pd3sd"); break;

50番台に put3&search を追加しました

(3) サンプルプログラム3について

サンプルプログラム3では、1020番ループを利用しながら条件を追加してきました。今回は「if - else if」を利用しましたが、配列やfor文、switch文、関数などを活用するともっとすっきりしたプログラムになるでしょう。ぜひ勉強を進めて、より強いクライアントにしてください。

14 疲労について

疲労はパラメータの組み合わせによって表現されています。動作系列では walk 系や put3 系など、系ごとに動作得点が決まっています。

また、アイテム類も「疲労」というイメージにあわせた得点になっています。

具体的な数値は以下の表のようになります。

動作得点		アイテム類得点	
walk系	-4	場外・white	-300
put2系	-100	穴	-9
put2&walk系	-10	ターゲット	50
kei系	-3	土	-20
put0系	-1	アイテム	10
put0&walk系	-2	warp系	0
dig系	-6	Client	0
put3系	-10	化石系	150
		三葉虫	100

15 その他

・グループについて

グループによって、ユーザ毎に利用できるルームとできないルームを分類しています。第一次予戦のルームや予戦練習用ルームは選手登録をしたユーザしかアクセスできませんので注意してください。

・サーバエラーについて

今年も第一次予戦の最終日付近ではサーバへのアクセスが集中し大変混雑することが予想されます。以前、混雑によって記録が残らないトラブルが発生しました。トラブルが起きた場合は結果の補正は行いませんのでご了承ください。

・接続人数制限について

負荷を軽減させるため、同時に接続できるユーザ数に制限を設けることとしました。制限人数についてはサーバの負荷状態によって調整しますので、接続できずに「User Error」で止まってしまう場合は、少し時間を置いてからアクセスをお願いいたします。

・自動巡回ユーザとの対戦方法について

自動巡回ユーザは各ルームをシェルスクリプトを利用して移動していきます。また、対戦が始まらなかった場合は一旦ゾンビ化し動きを停止してから次のルームへと移動します。

他ユーザのログイン動作をきっかけに次のルームへの移動を始めますので、

「同じルームに入ろうと思ったら逃げた」ように見えることがあります。クライアントプログラムを起動するときに、ルームオプション無しで起動するなどの工夫をするとスムーズに対戦することができます。

16 対戦案内

第一次予戦が次の要領で行われます。

日 時：2017年8月17日（木） 10：00ごろ から
8月30日（水） 19：00まで
※埼玉プロコンは7月31日（月）～8月10日（木） 19：00まで

ルーム：一人用ルームと二人用のルームが準備されます。
使用するルーム番号が決定しましたら、全情研ホームページで公開させていただきます。

二人用ルームでは必ず自動巡回ユーザを先攻として対戦してください。
自動巡回ユーザとの対戦方法は予戦練習ルームなどを利用して確実に後攻として接続できるようにしておいてください。
自動巡回ユーザが先攻になったのを確認するには「サーバ」「結果表示」から「対戦結果」で表示された一覧の「接続順」を確認してください。先攻が1、後攻が2となります。

自動巡回ユーザがどのルームにいるかは、
トップページの「待ち合わせ」「ログイン中のユーザ」で確認してください。

集 計：各クライアントが指定のルームで初めて取り組んだときの得点を集計します。

ユーザ ID：練習用 ID では第一次予戦ルームに入ることはできません。
選手用 ID を使って接続してください。
※事前に選手用 ID にて予戦練習ルームで動作確認をしておいてください。

注 意：ゲーム途中でクライアントが停止するなどのトラブルがあった場合は残りターン数に応じて減点されます。また、ルームへの接続回数も減点対象となります。
インターネット回線のトラブルの場合はゾンビ化解消後すぐに再度接続してください。どちらを採用するかは委員会にて検討させていただきます。

CHaserOnline のステップアップヒントは以上で終了となります。皆さんの参加をお待ちしています。