

```
1 #ifndef CODICIERRORI_H_
2
3 /*Le costanti seguenti contengono i valori degli errori
4 I codici di errori seguono la seguente codifica:
5 da -1000 a -2000 per i codici di errori relativi a
6 problemi tecnici
7 da -2000 a -3000 per i codici di errori relativi a
8 Registrazione e Log-In
9 da -3000 a -4000 per i codici di errori relativi a
10 errori su operazioni di Log
11
12 */
13 /*
14 Codici di errore relativi a problemi tecnici
15 */
16 #define ERR_NO_CONNECTION -1000
17 #define ERR_WRITE_ERROR -1001
18 #define ERR_INPUT_OUTPUT -1002
19 #define ERR_SOCKET_CREATION -1003
20 #define ERR_SOCKET_BINDING -1004
21 #define ERR_RECEIVING_MESSAGE -1010
22 #define ERR_SENDING_MESSAGE -1011
23 */
24 /*
25 Codici di errore relativi a problemi su Registrazione e
26 Log-In
27 */
28 #define ERR_INVALID_USERNAME -2000
29 #define ERR_NO_USER_FILE -2001
30 #define ERR_USERNAME_NOT_FOUND -2002
31 #define ERR_WRONG_PASSWORD -2003
32 #define ERR_INSERT_LOGGED_USER -2004
33 #define ERR_USER_ALREADY_LOGGED -2005
34
35 */
36 /*
37 Codici di errore relativi a problemi su operazioni di Log
38 */
39 #define ERR_LOG_FILE_CREATION -3000
40 #define ERR_LOG_WRITE -3001
41
42 #endif
```

```

1 #ifndef MESSAGGI_H_
2 #define MESSAGGI_H_
3
4 /*
5  * Stringhe per la generazione dei menu.
6  */
7 #define WELCOME_MESSAGE          "Benvenuto in Progetto
Laboratorio di Sistemi Operativi\nCosa vuoi fare?\nPremi 'l
' per eseguire il login.\nPremi 'r' per registrarti.\nPremi
'e' per uscire.\n"
8
9  /*
10  * Stringhe per la generazione di schermate di Login.
11  */
12 #define INSERT_USERNAME_LIM      "Inserisci il
tuo nome utente:\n"
13 #define INSERT_PASSWORD_LIM      "Inserisci la
tua password:\n"
14 #define SUCCESS_MESSAGE_LIM      "Bentornato!\n
Stai per essere reindirizzato ad una partita.\n"
15 #define WRONG_USERNAME_LIM       "Il nome
utente inserito non è stato trovato.\nInserisci di nuovo il
tuo nome utente:\n"
16 #define WRONG_PASSWORD_LIM       "Password
errata.\nInserisci di nuovo la tua password:\n"
17 #define USER_ALREADY_LOGGED      "L'utente
selezionato è già connesso.\nInserisci un nome utente
valido.\n"
18
19 /*
20  * Stringhe per la generazione della schermata di
Sign In.
21  */
22 #define INSERT_USERNAME_SIM       "Come vuoi
farti chiamare? (lunghezza massima 15)\n"
23 #define INSERT_PASSWORD_SIM       "Inserisci la
tua password (lunghezza massima 12).\n"
24 #define INSERT_PASSWORD_CONFIRM_SIM "Re-inserisci
la password.\n"
25 #define USER_ALREADY_PRESENT_SIM "Il nome
utente non è valido.\nScegli un altro nome, di lunghezza
massima 15, con cui farti chiamare:\n"
26 #define SUCCESS_MESSAGE_SIM       "Benvenuto!\n
Stai per essere reindirizzato ad un partita.\n"
27 #define GAME_SELECTION_MENU      "Cosa vuoi fare?\nPremi 'n'
per iniziare una nuova partita.\nPremi 'j' per unirti ad

```

```

27 una partita in corso.\nPremi 'e' per uscire.\n"
28 #define NO_SUCH_GAMES_MESSAGE      "Non ci sono partite
    disponibili in corso.\nPremi un tasto per continuare.\n"
29
30 //E' follia, follia dico però il messaggio non lo cancello
    perché l'ho scritto e mi scoccio troppo assai di
    cancellarlo
31 #define CURRENT_GAMES_PLAYED      "Partita %i.\nGiocatori
    : [...] \n"
32 #define SELECT_FROM_CURRENT      "Inserisci il numero della
    partita a cui vuoi aggiungerti.\n"
33 #define SELECT_DIRECTION          "Inserisci:\n\t'w' per
    muoverti verso NORD;\n\t'a' per muoverti verso OVEST;\n\t's
    ' per muoverti verso SUD\n\t'd' per muoverti verso EST"
34
35 /*
36  * Messaggi relativi al gioco.
37  */
38 #define VICTORY_MESSAGE            "\033[34;47m\t\t\tComplimenti
    hai vinto!\t\t\t\033[0m\n"
39 #define LOSS_MESSAGE              "\033[91;107m\t\t\tSei stato
    sconfitto.\t\t\t\033[0m\n"
40 #define COLLISION_MESSAGE         "Vedi che è occupata.\n"
41 #define TIME_OVER_MESSAGE         "\033[91mTEMPO SCADUTO\033[0m
    \n"
42 #define USER_LOG_OUT              "No Utente\n"
43 #define CLIENT_GONE               "MAI STATO UTENTE\n"
44 /*
45  * Messaggi di errore relativi al server.
46  */
47 #define NO_CONNECTION_ERR_MESSAGE "Impossibile
    connettersi al server\nL'applicazione verrà chiusa"
48 #define SOCKET_CREATION_ERR_MESSAGE "Impossibile
    creare la socket"
49 #define SOCKET_LISTEN_ERR_MESSAGE "Impossibile
    settare la socket in ascolto"
50 #define SOCKET_BINDING_ERR_MESSAGE "Impossibile
    collegarsi alla socket"
51 #define SOCKET_RECEIVE_DATA_ERR_MESSAGE "Impossibile
    ricevere dati dalla socket"
52 #define NOT_SURE_ERR_MESSAGE      "Qualcosa è
    andato storto"
53 #define ACCEPT_SOCKET_ERR_MESSAGE "Errore nella
    funzione accept"
54 #define USER_FILE_OPEN_ERR_MESSAGE "Errore nell'
    apertura del file RegisteredUser"

```

```

55 #define I_O_ERR_MESSAGE           "Errore di
    input output."
56 /*
57  * Messaggi di errore relativi al gioco stesso
58  */
59 #define FILLED_GAME_ERR_MESSAGE    "Non puoi unirti a
    questa partita..."
60 #define MATRIX_ALLOCATION_ERR_MESSAGE "Errore nella
    creazione della matrice di gioco"
61 #define ON_MOVEMENT_ERR_MESSAGE    "Errore nel
    movimento del giocatore"
62
63 /*
64  * Messaggi di errore lato client.
65  */
66 #define INVALID_CHOICE_ERR_MESSAGE "La scelta inserita
    non è valida.\n"
67
68 /*
69  * Messaggi di errore relativi ai thread.
70  */
71 #define THREAD_CREATION_ERR_MESSAGE "Impossibile creare
    un nuovo thread.\n"
72 /*
73  *Messaggi di LOG
74  */
75 #define LOG_START_SERVER           ": iniziata
    nuova sessione Server.\n"
76 #define LOG_CLOSE_SERVER           ": finita
    sessione Server.\n"
77 #define LOG_NEW_GAME               ": iniziata
    nuova partita con ID %d.\n"
78 #define LOG_END_GAME               ": finita
    partita con ID %d.\n"
79 #define LOG_NEW_PLAYER_JOIN         ": l'utente %s
    si è unito alla partita con ID %d.\n"
80 #define LOG_PLAYER_MOVES            ": in partita
    con ID %d l'utente %s è andato da %s a %s.\n"
81 #define LOG_PLAYER_WINS             ": l'utente %s
    ha vinto la partita con ID %d.\n"
82 #define LOG_NEW_USER               ": l'utente %s
    si è registrato al Server.\n"
83 #define LOG_SIGN_IN                 ": l'utente %s
    si è connesso al Server.\n"
84 #define LOG_SIGN_OUT                ": l'utente %s
    si è disconnesso dal Server.\n"

```

```
85 #define LOG_ERROR                                     ": ERRORE(%s).  
    \n"  
86 #define LOG_UNKNOWN_CLIENT_DISCONNECTION             ": un Client  
    non identificato con indirizzo ip %s si è disconnesso dal  
    Server.\n"  
87 #define LOG_PLAYER_TAKE                               ": in partita  
    con ID %d l'utente %s ha raccolto il pacco N %d da  
    posizione %s.\n"  
88 #define LOG_PLAYER_DELIVER                           ": in partita  
    con ID %d l'utente %s ha lasciato il pacco N %d in  
    posizione %s.\n"  
89 #define LOG_PLAYER_MAKE_POINT                        ": in partita  
    con ID %d l'utente %s ha fatto un punto.\n"  
90 #define IN_GAME_MENU "\033[30;47mW,A,S,D per muoverti Q  
    per raccogliere          E per consegnare\nR per  
    refresh                  0 per  
    uscire    \033[0m\n"  
91  
92  
93 #endif  
94
```

```

1  #ifndef CONSTANT_H_
2  #define CONSTANT_H_
3
4  #define MAX_PACCHI 16
5  #define MIN_OBSTACLES 5
6  #define MAX_SIZE_USERNAME 15
7  #define MAX_SIZE_PASSW 12
8  #define USERS_FILE "File/RegisteredUser.txt"
9  #define MAXIMUM_SOCKET_BACKLOG 10
10 #define clear() printf("\033[H\033[J");
11 #define leggi() n_b_r=read(sockfd,msg,6);msg[n_b_r]='\0';
    n_b_r=atoi(msg);write(sockfd,"1",strlen("1")) ;n_b_r=read(
    sockfd,msg,n_b_r);msg[n_b_r]='\0';printf("%s",msg);
12
13 /* Le costanti seguenti contengono valori utili alla
    gestione della matrice di
14 * gioco.
15 */
16 #define RESET "\033[0m"
17 #define GREEN "\033[92m"
18 #define RED "\033[91m"
19 #define CYAN "\033[96m"
20 #define YELLOW "\033[93m"
21 #define BG_WHITE_FG_BLACK "\u001b[47;30m"
22
23 /* Le costanti seguenti contengono valori utili per gestire
    una partita.
24 *
25 */
26 #define MAX_GAMES_PLAYABLE 20
27 #define MAX_PLAYER_N 8
28 #define MAX_OBSTACLES_N 20
29 #define MAX_GRID_SIZE_L 15
30 #define MAX_GRID_SIZE_H 10
31 #define MAX_TIME 60
32 #define PLAYER_EXITS 1000
33 #define GAME_END_FOR_TIME 1001
34 #define PLAYER_MAKE_A_POINT 1002
35 /*
36 La struttura seguente contiene i dati utili per la
gestione di un giocatore all'interno di una Partita:
37 -(posx, posy) rappresentano la sua posizione all'
interno della griglia di gioco;
38 -pacco è un valore booleano che rappresenta se il
giocatore ha o meno un pacco in carico;
39 -codicePacco è un intero a quattro bit, se pacco è

```

```

39 impostato ad 1 rappresenta l'identificativo del pacco
   stesso;
40 -nome è una stringa contenente l'username del giocatore;
41 */
42 typedef struct {
43     int posX;
44     int posY;
45     unsigned int pacco:1;
46     unsigned int codicePacco:4;
47     char nome[MAX_SIZE_USERNAME];
48 } player;
49
50 typedef struct {
51     char user[MAX_PLAYER_N][MAX_SIZE_USERNAME];
52     pthread_mutex_t sem;
53 } loggedUser;
54
55 /*
56 La struttura GameGrid rappresenta la singola cella della
   matrice di gioco, contiene 23 bit suddivisi in vari campi:
57 - ostacolo, 1 bit, è un flag booleano. Vale 1 se la
   cella contiene un ostacolo, 0 altrimenti;
58 - giocatore, 1 bit, è un flag booleano. Vale 1 se la
   cella contiene un giocatore, 0 altrimenti;
59 - pacco, 1 bit, è un flag booleano. Vale 1 se la cella
   contiene un pacco, 0 altrimenti;
60 - locazione, 1 bit, è un flag booleano. Vale 1 se la
   cella contiene una locazione di consegna, 0 altrimenti;
61 - codiceGiocatore, 3 bit, è un intero. Contiene un
   numero da 0 a 7, identificativo di uno specifico giocatore
   . Se il flag giocatore
62 è settato ad 1 viene utilizzato per rappresentare
   uno specifico giocatore sulla matrice.
63 - codicePacco, 4 bit, è un intero. Viene utilizzato per
   rappresentare uno dei 16 pacchi disponibili all'interno
   della partita;
64 - codiceLocazione, 4 bit, è un intero. Viene utilizzato
   per rappresentare una delle 16 locazioni disponibili.
   Contiene lo stesso valore
65 del pacco associato a quella locazione;
66 - I restanti 8 bit vengono usati come codifica one hot
   della visibilità. Ogni qual volta un giocatore scopre una
   casella il bit corrispondente
67 al suo codiceGiocatore.
68 */
69 typedef struct {

```

```

70  unsigned int ostacolo:1;
71  unsigned int giocatore:1;
72  unsigned int pacco:1;
73  unsigned int locazione:1;
74  unsigned int codiceGiocatore:3;
75  unsigned int codicePacco:4;
76  unsigned int codiceLocazione:4;
77  unsigned int permessi[MAX_PLAYER_N];
78 } GameGrid;
79
80 /*
81  La struttura Game contiene le informazioni necessarie a
    gestire una partita:
82      -gameId, un intero contiene l'id della partita
    corrispondente al pid del processo che l'ha in gestione;
83      -grid, contiene la matrice di gioco;
84      -giocatori, un array di struct player contente le
    informazioni sui giocatori della partita. L'id del
    giocatore corrisponde alla
85      sua indicizzazione nell'array;
86      -piena, un flag booleano. Vale 1 se alla partita non
    si possono aggiungere altri giocatori, 0 altrimenti;
87      -sem, un semaforo associato alla partita, serve a far
    sì che solo un client per volta possa accedere alla
    matrice di gioco;
88      -punteggio, un array di interi. Ad ogni casella e
    associato un giocatore tramite il suo id e contiene il
    numero di pacchi consegnati
89      per ogni giocatore;
90      -i due array di interi locazioneXPacchi e
    locazioneYPacchi contengono informazioni su dove ogni
    singolo pacco vada consegnato.
91  */
92  typedef struct {
93      int gameId;
94      GameGrid **grid;
95      player giocatori[MAX_PLAYER_N];
96      unsigned int piena:1;
97      pthread_mutex_t sem;
98      unsigned int timeOver : 1;
99      int punteggio[MAX_PLAYER_N];
100     int locazioneXPacchi[MAX_PACCHI];
101     int locazioneYPacchi[MAX_PACCHI];
102 } Game;
103
104 typedef struct {

```



```
105  int fd;
106  pthread_mutex_t sem;
107 } LogFile;
108
109 #endif
110
```