
1 Servidor GS - Organização de dados

O servidor GS instalado na máquina ‘tejo’ do laboratório LT5 usa uma estrutura de directorias para armazenar toda a informação referente ao protocolo de forma persistente entre sessões.

Os alunos podem replicar total ou parcialmente a estrutura de directorias aqui descrita, a qual foi concebida tendo em vista a simplificação do processamento para armazenamento flexível e indexação de jogos e *scores*, aproveitando ainda as estruturas de dados previstas no *filesystem* do Linux para obter facilmente a ordenação de ficheiros nele contidos.

Na directoria de trabalho do servidor GS encontram-se duas directorias designadas GAMES e SCORES.

Assume-se que no arranque do servidor GS estas duas directorias já estão criadas na directoria de trabalho do GS, bem como o ficheiro que contém as palavras a decifrar e os ficheiros de sugestões a enviar ao player sob solicitação do mesmo.

1.1 A directoria corrente

A directoria corrente do GS contém o ficheiro com as palavras a decifrar (ficheiro texto), as directorias GAMES e SCORES, e os ficheiros de sugestões. O ficheiro de palavras a decifrar obedece ao seguinte formato:

palavra(1) ficheiro_hint(1)

palavra(2) ficheiro_hint(2)

..

palavra(n) ficheiro_hint(n)

Cada linha deste ficheiro contém em primeiro lugar uma palavra a decifrar pelos *players* seguida do nome do correspondente ficheiro de sugestão, o qual será enviado ao player caso este execute o comando 'hint'. O servidor GS no tejo selecciona em sequência as palavras a serem decifradas à medida que vai recebendo mensagens SNG dos diferentes players.

1.2 A directoria GAMES

A directoria GAMES contém o estado dos jogos em curso e o resumo dos jogos já terminados.

O estado de um jogo em curso para um dado PLID encontra-se guardado num ficheiro designado GAME_(plid).txt

Os jogos já terminados pertencentes a um dado PLID ficam guardados, para cada player, numa directoria cuja designação é PLID sob a directoria GAMES. O caminho para a localização dos ficheiros de jogos finalizados é assim: GAMES/PLID.

O ficheiro GAME_(plid).txt tem o seguinte formato:

palavra ficheiro_hint

code(1) play(1)

code(2) play(2)

.. ..

code(n) play(n)

A primeira linha contém a palavra a decifrar que foi seleccionada pelo GS para o jogo em curso quando recebeu a mensagem SNG, e a designação do ficheiro de ajuda a enviar para o player quando este emitir o comando 'hint'.

A partir da segunda linha inclusivé, cada linha do ficheiro refere-se a uma jogada. Cada jogada é caracterizada pela variável 'code' e pela

jogada 'play'. Quando a variável 'code' tem o valor T (Trial) significa que 'play' é uma letra enviada ao GS pelo player na sequência de um comando 'play'. Quando a variável 'code' tem o valor G (Guess) significa que 'play' é uma palavra enviada ao GS pelo player na sequência de um comando 'guess'.

Apenas as jogadas com um novo conteúdo são armazenadas neste ficheiro. A uma jogada com conteúdo repetido e um número de jogada igual ao esperado pelo GS este responde com status==DUP não registando a jogada no ficheiro GAME_(plid).txt. No entanto, se após a última jogada armazenada no ficheiro o GS receber uma jogada com o mesmo conteúdo e o mesmo número dessa última jogada armazenada, o GS interpretará esta jogada como sendo uma repetição de jogada anterior para a qual o *player* não recebeu a resposta. Nesse caso o GS repetirá a resposta.

Caso a jogada recebida pelo GS não tenha o número de jogada esperado pelo GS este responderá com INV. Caso a jogada recebida pelo GS tenha o mesmo número da última jogada registada, mas não tenha o mesmo conteúdo, o GS responderá com status==INV.

Quando um jogo termina, ele pode terminar com sucesso (WIN), insucesso (FAIL) ou desistência (QUIT). Quando um jogo termina, o ficheiro GAME_(plid).txt é transferido para a directoria do player em causa GAMES/PLID a qual é criada automaticamente quando termina o primeiro jogo do referido player.

Dentro da directoria GAMES/PLID os jogos finalizados pelo player PLID são guardados com uma nova designação que facilita o seu tratamento em posteriores operações de ordenação.

Ao ser transferido para a directoria GAMES/PLID, o ficheiro GAME_(plid).txt passa a ter uma designação com o formato:

YYYYMMDD_HHMMSS_(code).txt

Em que YYYYMMDD é a data de finalização do jogo, HHMMSS é a hora de finalização do jogo e code é uma letra de três possíveis (W, F ou Q) e que indica como foi finalizado o jogo - por Win, Fail ou Quit, respectivamente.

Exemplo:

O player com o PLID=123456 executou o comando *start* 123456. Ao receber SNG correspondente ao comando *start*, o servidor criou o

ficheiro GAME_123456.txt

Se após as jogadas o jogo terminou com sucesso às 20 horas, 10 minutos e 5 segundos do dia 21 de Novembro de 2022, o ficheiro GAME_123456.txt é transferido para a directoria GAMES/123456 (sendo esta criada caso não exista ainda). Ao ser transferido para a directoria 123456, o ficheiro GAME_123456.txt assume a designação 20221121_201005_W.txt.

Esta metodologia de designação dos jogos já terminados vai permitir ao servidor ordenar facilmente os jogos já terminados por ordem decrescente de data de finalização. Assim o jogo mais recente vai figurar no topo da lista.

1.3 Comandos de eliminação de dados

Durante o processo de desenvolvimento da aplicação *player* os alunos podem solicitar ao GS no tejo a eliminação de ficheiros GAME_(plid).txt e de directorias GAMES/PLID usando o comando do Linux:

```
nc -u tejo(.tecnico.ulisboa.pt) 58011
```

Após o comando nc, para eliminar o ficheiro GAME_(plid).txt, deve ser inserida a linha:

```
KILLGAME PLID (seguida de return)
```

Após o comando nc, para eliminar a directoria GAMES/PLID, deve ser inserida a linha:

```
KILLPDIR PLID (seguida de return)
```

O servidor GS executa as operações acima referidas sem responder.

1.4 A directoria SCORES

Quando um dado jogo termina com sucesso, é também criado um ficheiro de 'score' na directoria SCORES

A designação do ficheiro de 'score' é:

score_PLID_DDMMYYYY_HHMMSS.txt

em que score tem um valor [001 - 100] sempre com três dígitos e PLID é a identificação do jogador. DDMMYYYY e HHMMSS são a data e a hora a que o jogo terminou com sucesso respectivamente.

A designação acima facilita a obtenção da lista de 'scores' ordenados por valor decrescente de 'score'.

O ficheiro scores tem uma única linha com o seguinte formato:

score PLID palavra n_succ n_trials

em que score é a pontuação obtida pelo GS como percentagem de tentativas bem sucedidas sobre o total de tentativas, e n_succ e n_trials são o número de tentativas bem sucedidas e o número total de tentativas respectivamente.

1.5 Ordenação de ficheiros em directorias por nome

Listas ordenadas de ficheiros de jogos finalizados dentro de cada directoria de PLID podem ser obtidas recorrendo à função scandir().

A função FindLastGame() que se lista abaixo, é usada pelo GS em execução no tejo para obter o ficheiro de jogo mais recente para um dado PLID. O nome do ficheiro é retornado para a variável apontada por fname.

```
int FindLastGame(char *PLID, char *fname)
{
    struct dirent **filelist;
    int n_entries, found;
    char dirname[20];

    sprintf(dirname, "GAMES/%s/", PLID);
    n_entries = scandir(dirname, &filelist, 0, alphasort);
    found=0;
```

```

if (n_entries <= 0)
    return(0);
else
{
    while (n_entries --)
    {
        if( filelist [ n_entries ]->d_name[0] != ' . ' )
        {
            sprintf(fname,"GAMES/%s/%s",PLID, filelist [ n_entries ]->d_name);
            found=1;
        }
        free( filelist [ n_entries ] );
        if(found)
            break;
    }
    free( filelist );
}
return(found);
}

```

A função FindTopScores() listada abaixo é usada pelo GS em execução no tejo para ter acesso a cada um dos 10 ficheiros com scores mais elevados contidos na directoria SCORES.

```

int FindTopScores(SCORELIST *list)

```

```

{
    struct dirent **filelist;
    int n_entries, i_file;
    char fname[50];
    FILE *fp;

    n_entries = scandir("SCORES/", &filelist, 0, alphasort);

    i_file=0;
    if (n_entries < 0)
    {
        return(0);
    }
    else
    {
        while (n_entries--)
        {
            if(filelist[n_entries]->d_name[0]!='.')
            {
                sprintf(fname,"SCORES/%s",filelist[n_entries]->d_name);
                fp=fopen(fname,"r");
                if (fp!=NULL)
                {
                    fscanf(fp,"%d_%s_%s_%d_%d",

```

```

        &list->score[i_file], list->PLID[i_file], list->word[i_file], &list->n_succ[i_file],
        &list->n_tot[i_file]);
    fclose(fp);
    ++i_file;
}
}

free(filelist[n_entries]);
if(i_file==10)
    break;
}
free(filelist);
}

list->n_scores=i_file;
return(i_file);
}

```

A definição das estruturas relativas a directorias nas funções acima ilustradas estão contidas no ficheiro ‘dirent.h’.

1.6 Envio de ficheiros solicitados pelos comandos *state* e *scoreboard*

Os ficheiros de texto que o GS envia para o *player* na sequência da recepção de uma mensagem STA ou GSB já se encontram formatados para apresentação ao utilizador. O *player* apenas tem de ler o ficheiro que recebeu e gravou na sua directoria de trabalho (com fgets()) por

exemplo) e imprimir sequencialmente no écran todas as suas linhas. Cada linha vem já terminada com ‘\n’ pelo GS.

Os ficheiros resultantes da emissão do comando *state* baseiam-se em ficheiros de jogo com formato único (já descrito), quer o jogo esteja activo quer esteja finalizado. No entanto, o formato do ficheiro enviado pelo GS em resposta a uma mensagem STA terá um formato se o jogo ainda estiver activo diferente do formato que terá se o jogo estiver finalizado.

Os ficheiros enviados pelo GS no tejo em resposta a uma mensagem STA, têm a designação STATE_(plid).txt.

Os ficheiros enviados pelo GS no tejo em resposta a uma mensagem GSB, têm a designação TOPSCORES_nnnnnnnn.txt em que nnnnnnnn é o *process ID* do processo que gerou o ficheiro no GS.