Rapor
FINAL

Make

Usage ./mine direct1 direct2 ….. directN

Usage : ./miner direcNum(integer)

Program once make yapılır daha sonra sekilde gösterildigi gibi calıştırılır. Mine calıştıktan sonra miner I ları bekler ve CTRL-C yapıncaya kadar calışır .

Miner

Mine a baglanarak ordan n tane directory alıp icindeki kelimeleri bulup karşılıgında para alır eger icinde direct varsa mine a tekrar yollar ve kendi log file I na yazar.

```
typedef struct server

{

        int sizeS;

        char cDirectory[BUFSIZE][BUFSIZE];


}server;


typedef struct client

{

        int size;

        int pid;

}client;

typedef struct words
```

```
{
	char **words;
	int *wordsSize;
	int sizeW;
}words;

typedef struct allWords
{
	/* data */
	char words[BUFSIZE][BUFSIZE];
	int wordsSize[BUFSIZE];
	int size;
}allWords;


typedef struct points
{
	int point[BUFSIZE];
	int size;
	int isSent;
}points;




void handle( int sig);
```

```
int main2(char *directory);
/*Takes file directory and return integer */
int countWords(char *fileDir);


int addWordsFile(char **words,int arraySizeX);

/*it takes fileName char* */
/* return max line boyut  */
int getBoyut(char *file);

/*it takes filename char* */
/* return toplam satir sayisi */
int getLineNum(char *file);

/*it takes char ** and arraySizeX return void  */
/* verilen iki boyutlu array i free yapar  */
void freeArray(char **arr, int arraysSizeX);

/*is takes two size as a integer and return char ** */
/* verilen iki boyutlu array a size lar kadar yer alir */
char** Make2DintArray(int arraySizeX, int arraySizeY);

/*is takes fileName and char ** and arraySize as a integer return void */
/* verilen array a dosyadan okuma yapar  */
void readData(char *file,char **arr,int arraySizeX);
```

```
/*it takes array and it sizex return total words inside*/

int countWordsInArray(char **arr,int arraySizeX);



/*it takes array and return all words in it */

char ** readWords(char **arr,int arraySizeX,int wordCount);



/*it is reading temp file and adding words array after that delete temp file return set words size*/

int readWordsFile(char **words,int *wordsCount);



/*it is checking a word is in words array if is contain return its index else return -1*/

int isContain(char **words,char *word,int arraySizeX);



void* threadOperation(void* data);

char** Make2DintArray(int arraySizeX, int arraySizeY);

void freeArray(char **arr, int arraysSizeX);

int client2();
```

Mine

Mine ise kendisine gelen miner a lara directory verir ve getirdikleri kelime karsılıgında para verir.

```
typedef struct server
{
        /* data */
        char cDirectory[BUFSIZE][BUFSIZE];
        int size;
}server;
```

```
typedef struct client
{
	int size;
	int pid;
}client;

typedef struct words
{
	char **words;
	int *wordsSize;
	int sizeW;
}words;

typedef struct allWords
{
	/* data */
	char words[BUFSIZE][BUFSIZE];
	int wordsSize[BUFSIZE];
	int size;
}allWords;

typedef struct points
{
	int point[BUFSIZE];
	int size;
	int isSent;
}points;
typedef struct Miner
{
	double  startTime;
	double stopTime;
	int price;
	int pid;

}Miner;

typedef struct Miners
{
	Miner workingMiners[BUFSIZE];
	Miner pendingMiners[BUFSIZE];
	Miner servedMiners[BUFSIZE];
	int sizeWorking;
	int sizePending;
	int sizeServed;
}Miners;



int isCreateClient(int pid1,int pid2);
/*CTRL-C handler */
void handle( int sig);
void signalPipeHandle(int signo);/*Pipe handler*/
void my_sigchld_handler(int sig);/*child process handler */
void  ALARMhandler(int sig);

/*gives operation name number size and client fifoname and calculate it sent client*/
void calculateOperatipn(char fifoName1[],char *operation,int * numbers,int size);

/*my server takes pipe parameter and fifo name connect client */


/*my sleep dies after 30 seconds if nothing connected */
void mySleep(clock_t sec);

/*verilen kelime icindeyse index I degilse -1 yollar */
int isInside(char *word);


char** Make2DintArray(int arraySizeX, int arraySizeY);
void freeArray(char **arr, int arraysSizeX);
void*  server2(void *ignoreMe);
```