

CSI 333 – Programming at the Hardware-Software Interface

Illustrating the use of fgets and fputs

Handout 7.1

Description: The executable version of this program should be run using a command line of the form

```
findpat infile outfile pattern
```

The program repeatedly reads a line from an input file (specified by the command line argument `argv[1]`), checks whether the line contains a given pattern (specified by the command line argument `argv[3]`), and if so, writes the line to an output file (specified by the command line argument `argv[2]`). This is repeated until there are no more lines in the input file.

```
#include <stdio.h>
#include <stdlib.h> /* Allows us to use the exit function. */
#include <string.h> /* Allows us to use the strstr function. */

/* Maximum line length. */
#define MAXLEN 81

/* Symbolic constants associated with command line parameters.*/
#define NUMARG 4
#define IN_FILE_ARG 1
#define OUT_FILE_ARG 2
#define PATTERN_ARG 3

int main(int argc, char *argv[]){
    /* Each input line is assumed to have at most MAXLEN-1 characters */
    /* including the newline character. */

    char line[MAXLEN]; /* To hold each line of the input file. */
    FILE *finp, *foutp; /* Pointers for input and output files. */

    /* The command line must specify the input file, the output file */
    /* and the pattern -- a total of NUMARG arguments. */
    if (argc != NUMARG) {
        printf("Usage: findpat infile outfile pattern\n"); exit(1);
    }

    /* Strings argv[1] and argv[2] are assumed to specify */
    /* the names of the input and output files respectively. */

    /* Open the input file for reading. */
    if ((finp = fopen(argv[IN_FILE_ARG], "r")) == NULL) {
        /* Open failed. */
        printf("Could not open file %s for reading.\n", argv[IN_FILE_ARG]); exit(1);
    }
}
```

(over)

```

/* Open the output file for writing. */
if ((foutp = fopen(argv[OUT_FILE_ARG], "w")) == NULL) {
    /* Open failed. */
    printf("Could not open file %s for writing.\n", argv[OUT_FILE_ARG]); exit(1);
}
/* Both the files have been opened successfully. */

/* The pattern is given by the string argv[3]. Read one line at a time */
/* from the input file, check if the pattern occurs in the line, and */
/* if so, write the line to the output file. */

/* The string library function strstr is used for checking the */
/* occurrence of pattern in the line that has been read in. */

while (fgets(line, MAXLEN, finp) != NULL) {
    if (strstr(line, argv[PATTERN_ARG]) != NULL) { /* Successful match. */
        fputs(line, foutp);
    }
}

/* Close the files and stop. */

if (fclose(finp) == EOF) { /* Error in closing input file */
    printf("Error in closing file input.dat.\n");
}

if (fclose(foutp) == EOF) { /* Error in closing output file */
    printf("Error in closing file output.dat.\n");
}
return 0;
} /* End of main. */

```