

CSI 333 – Programming at the Hardware-Software Interface

Handout 6.4

The following program is similar to the program discussed in Handout 6.2. The only difference is that this program reads its input from a file called `input.dat`. It counts the number of lines, words and characters in the input file and prints the results to `stdout`.

You should note the way the file is opened for reading, the use of the library function `getc` to read characters from the input file and the way file is explicitly closed in the program.

The input file name is hardcoded into the program. We will see later how the program can be written so that the file name can be specified at run time.

```
#include <stdio.h>
#define YES 1
#define NO 0

int main(void) {

    char *infile = "input.dat"; /* Input file name. */
    FILE *finp; /* File pointer for input file. */

    int c; /* To read chars from input file. */
    int nl, nw, nc; /* To count number of lines, words and chars. */
    int inword; /* To remember whether we are within a word. */

    /* Open the input file for reading. */

    if ((finp = fopen(infile, "r")) == NULL) {
        /* Open failed. */
        printf("Could not open file input.dat for reading.\n");
        exit(1);
    }

    /* File opened successfully. Now, count words, lines */
    /* and characters. */

    inword = NO; nl = nw = nc = 0;

    while ((c = getc(finp)) != EOF) { /* Note the use of getc. */
        ++nc; /* One more new character read in. */
        if (c == '\n')
            ++nl; /* One more line of input. */

        if ((c == ' ') || (c == '\n') || (c == '\t')) {
            /* White space characters terminate a word. */
            inword = NO;
        }
    }
}
```

```
    }
    else {
        /* A non-whitespace char seen. A new word begins if */
        /* we are not already within a word.                */
        if (inword == NO) {
            inword = YES; ++nw;
        }
    }
} /* End of while. */

printf("%d %d %d \n", nl, nw, nc); /* Output results to stdout. */

/* We are done with the input file. We should close it. */

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

return 0;
} /* End of main. */
```