

Re: Write to file

Source: http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2007-06/msg01860.html

- *From:* Ben Bacarisse <ben.usenet@xxxxxxxxx>
 - *Date:* Wed, 13 Jun 2007 13:40:36 +0100
-

Bill <bill.warner@xxxxxxxxxxxxx> writes:

Thanks alot. I'm making progress. No errors now, but the output in the file is screwy. The code:

Please trip your replies (and always trim signature lines).

```
#include <stdio.h>
#include <ctype.h>
#include <string.h>

#define DISPLAY 117 /* Length of display line */
#define PAGE_LENGTH 20 /* Lines per page */

int main(int argc, char *argv[])
{
    FILE *pfile; /* File pointer */
    FILE *outfile;
    char *p;
    int c;

    unsigned char buffer[DISPLAY/4 - 1]; /* File input buffer */
    int count = 0; /* Count of characters in buffer */
    int lines = 0; /* Number of lines displayed */
    int i = 0; /* Loop counter */

    char string [100];
    pfile = fopen ("my.txt" , "r");

    while((c = fgetc(pfile)) != EOF) /* Continue until end of
file */
    {
        if(count < sizeof buffer) /* If the buffer is not full */
        /* Read a character */
        buffer[count++] = (unsigned char)fgetc(pfile);
```

Re: Write to file

I intended you use c here not read another character!

```
else
{
/* Now display buffer contents as characters */
for(count = 0; count < sizeof buffer; count++)
printf("%c", isprint(buffer[count]) ? buffer[count]:'.');
printf("\n"); /* End the line */
count = 0; /* Reset count */
```

Your input is long enough to trigger this condition. You reuse the buffer.

```
if(!(++lines%PAGE_LENGTH)) /* End of page? */
if(getchar()=='E') /* Wait for Enter */
return 0; /* E pressed */
}
}
```

```
/* Display last line as characters */
for(i = 0; i < count; i++)
{
printf("%c",isprint(buffer[i]) ? buffer[i]:'.');
}
}
```

```
printf("\n");
fclose(pfile); /* Close the file */
```

```
//re-open file and write out the buffer contents
outfile = fopen ("myOutput.txt" , "w");
```

```
if(outfile==NULL)
{
return 0;
}
}
```

```
for(count = 0; count < sizeof buffer; count++)
{
if(!isprint(buffer[count]))
{
buffer[count] = '.';
}
else
{
fputc(buffer[count],outfile);
}
}
```

Re: Write to file

You took a suggestion to modify the buffer and combined it with single character printing to produce something rather odd. It is not wrong, just odd. Why change `buffer[count]` when is it not a printing character if you don't then print it?

```
//fputc(outfile, isprint(buffer[count]) ? buffer[count]:'.');
}

/*for(count = 0; count < sizeof buffer; count++)
{
(int) fwrite(buffer, 1, isprint(buffer[count]), outfile);
}*/

fclose(outfile);
getchar();
return 0;
}
```

Input file contents:
Fred Flinstone 123-456-7890
Barney Rubble 123-789-4561

Output file:
B12y7p456r

Not what I get, but still.

I should point out that although I have only commented on details, the overall design of this program looks a bit wacky. If you intend to process text files, why do you have a buffer of size $117/4-1$? Why are you happy to simply reuse the space when you read more than these 28 characters? If the input has 30 characters, your buffer will contain numbers 29 and 30 followed by character numbers 3, 4 and so on. It all looks very odd.

You should maybe say what you are actually trying to do.

—

Ben.

.