



System Programming

Lab 5

Stdio Compare

Lecturer : Professor Pao-Ann Hsiung

Teaching Assistants: 湯凱鈞 & Adarsh

Embedded Systems Laboratory
National Chung Cheng University
Chiayi, Taiwan-62102

[Outline

- Stdio Compare
- Requirements
- Turn In
- Download



Stdio Compare

[Figure 5.4 Use getc & putc]

```
#include      "apue.h"

int
main(void)
{
    int      c;

    while ( (c = getc(stdin)) != EOF)
        if (putc(c, stdout) == EOF)
            err_sys("output error");

    if (ferror(stdin))
        err_sys("input error");

    exit(0);
}
```

[Char-at-a-time : getc & putc]

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

- `int getc(FILE *fp);`

Return: **next char** if OK, EOF on end of file or error.

- `int putc(int c, FILE *fp);`

Return: `c` if OK, EOF on error

[Figure 5.5 Use fgets & fputs]

```
#include      "apue.h"

int
main(void)
{
    char      buf[MAXLINE] ;

    while (fgets(buf, MAXLINE, stdin) != NULL)
        if (fputs(buf, stdout) == EOF)
            err_sys("output error");

    if (ferror(stdin))
        err_sys("input error");

    exit(0);
}
```

[Line-at-a-Time : fgets & fputs]

#include <stdio.h>

■ char *fgets(char *buf, int n, FILE *fp);

- n : the maximum number of characters to be read
- Stores : line + **\n** + NULL
- Return: buf if OK, NULL on EOF or error

■ int fputs(const char *str, FILE *fp);

- Need to put an extra **\n** before writing out as a line
- Return : nonnegative value if OK, EOF on error

[setvbuf]

- `int setvbuf(FILE *fp, char *buf, int mode, size_t size);`
- `buf` : user allocated buffer
- If set to NULL, the function automatically allocates a buffer of the specified size.
 - Files: `st_blksize` in `stat`
 - Pipes: `BUFSIZ` (`stdio.h`)

int mode	
<code>_IOFBF</code>	fully buffered
<code>_IOLBF</code>	line buffered
<code>_IONBF</code>	unbuffered

- `size_t size` : buffer size

A decorative graphic consisting of a thin gold circle on the left and a horizontal bar extending to the right. The bar has a gold-to-white gradient. A large black left square bracket is on the left, and a large gold right square bracket is on the right.

Requirements

[Requirements Part 1]

- Measure the execution time durations and count the loop iterations in Figure 5.4 / Figure 5.5, by modifying the given program.
- Use the `dd` command to create a 100MB file
- Add a `counter` into Figure 5.4 / Figure 5.5
- **Hint :** Use `time` command to measure the execution time.
- **Hint:** `dd if=/dev/zero of=100mb_files ...`

[Requirements Part 1 (cont'd)]

- Complete the table1

Function	User CPU	System CPU	Real time	Loop Iterations
getc , putc				
fgets , fputs				

[Requirements Part 2]

- Change the buffer mode in Figure 5.5 and measure the execution time
 - Hint : Use `setvbuf` to change the buffer mode.
 - Hint : Use `time` command to measure the execution time.

[Requirements Part 2 (cont'd)]

- Complete the table2

Mode	User CPU	System CPU	Real time
Fully buffered			
Line buffered			
unbuffered			

[Result of Requirements]

- Submit the following in table format (Answer.doc):
 - ✓ Where is the counter you added?
 - ✓ Screen shot for Table 1
 - ✓ Screen shot for Table 2



Turn In

[Turn In]

- The E-course2
 - <https://ecourse2.ccu.edu.tw/>
- Upload 學號.zip into “Lab_5”
 - Source files
 - Answers.pdf
 - LAB5_Tables.pdf
- Due date
 - 2023/05/11 23:59:59 遲交*0.8 超過一週*0.6

[Turn In (cont'd)]

- TA's email:
 - 湯凱鈞 : 4685231GF@gmail.com
 - Adarsh: vtu10666@veltechuniv.edu.in