

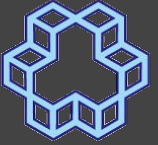


# Fundamentals of Programming

## session 31

### File Processing

```
*****  
* convolve.c  
***** /  
  
/* Standard includes */  
#include <assert.h>  
#include <math.h>  
#include <stdlib.h> /* malloc(), realloc() */  
  
/* Our includes */  
#include "base.h"  
#include "error.h"  
#include "convolve.h"  
#include "klt_util.h" /* printing */  
  
#define MAX_KERNEL_WIDTH 71  
  
typedef struct {  
    int rdd;  
    float data[MAX_KERNEL_WIDTH];  
} ConvolutionKernel;  
  
/* Kernels */
```

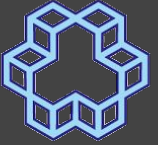


1928

K. N. Toosi University of Technology

# Why files?

- Program data (variables, arrays, structures, etc.) is stored in ram
  - removed once the program finishes
- Secondary storage
  - Hard Disk
  - SSD
  - Flash disk
  - CD/DVD
  - :



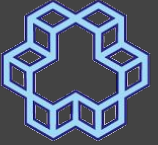
1926

K. N. Toosi University of Technology

# Remember: Streams

- A sequence of data (bytes)
  - processed sequentially
  - potentially unlimited
- Stream vs. Batch data
- user input, user output, files, I/O devices, etc.

# Remember: Stream Processing



1926

K. J. Somaiya Institute of Technology

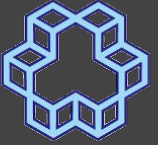
423423234532asdfsadfasdfzxc...

```
while (1) {  
    char c = getchar();  
    putchar(toupper(c));  
}
```

423423234532ASDFSADFSDFZXC...

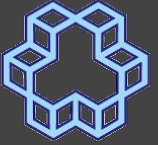
# Remember: Standard streams

- **Standard Input**
- **Standard Output**
- **Standard Error**



1926

K. J. Somaiya Institute of Technology



1926

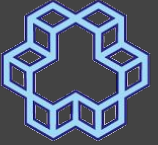
K. J. Somaiya Institute of Technology

# Remember: Standard streams

- Standard Input
- Standard Output
- Standard Error

`stdio.h`

```
extern struct _IO_FILE *stdin;  
extern struct _IO_FILE *stdout;  
extern struct _IO_FILE *stderr;
```



1926

K. N. Toor University of Technology

# Files as streams

- Standard Input
- Standard Output
- Standard Error

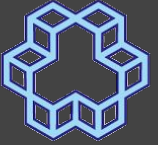
`stdio.h`

```
extern struct _IO_FILE *stdin;  
extern struct _IO_FILE *stdout;  
extern struct _IO_FILE *stderr;
```

## Open a file -> stream

- Read from stream
- Write to stream

```
struct _IO_FILE;  
typedef struct _IO_FILE FILE;
```



1926

K. N. Toosi University of Technology

# Writing to streams

```
extern struct _IO_FILE *stdin;  
extern struct _IO_FILE *stdout;  
extern struct _IO_FILE *stderr;
```

```
struct _IO_FILE;  
  
typedef struct _IO_FILE FILE;
```

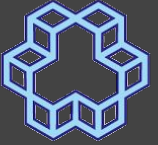
## Open a file -> stream

- Read from stream
- Write to stream

```
int printf(const char *format, ...);  
int fprintf(FILE *stream, const char *format, ...);
```

```
printf("Salam Kako!!!!\n");  
fprintf(stdout, "Salam Kako 2!!!!\n");  
fprintf(stderr, "Salam Kako 3!!!!\n");  
  
FILE *f;  
  
f = stdout;  
  
fprintf(f, "Salam Kako 4!!!!\n");
```





1926

K. N. Toosi University of Technology

# Writing to streams

```
extern struct _IO_FILE *stdin;  
extern struct _IO_FILE *stdout;  
extern struct _IO_FILE *stderr;
```

```
struct _IO_FILE;  
  
typedef struct _IO_FILE FILE;
```

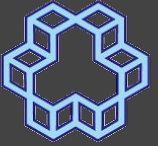
## Open a file -> stream

- Read from stream
- Write to stream

```
int printf(const char *format, ...);  
int fprintf(FILE *stream, const char *format, ...);
```

```
printf("Salam Kako!!!!\n");  
fprintf(stdout, "Salam Kako 2!!!!\n");  
fprintf(stderr, "Salam Kako 3!!!!\n");  
  
FILE *f;  
  
f = stdout;  
  
fprintf(f, "Salam Kako 4!!!!\n");
```

**Never use `_IO_FILE`!**



1926

K. N. Toopi University of Technology

# Writing to streams

```
extern struct _IO_FILE *stdin;  
extern struct _IO_FILE *stdout;  
extern struct _IO_FILE *stderr;
```

```
struct _IO_FILE;
```

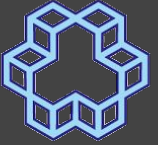
```
typedef struct _IO_FILE FILE;
```

```
printf("Salam Kako!!!!\n");  
  
fprintf(stdout, "Salam Kako 2!!!!\n");  
  
fprintf(stderr, "Salam Kako 3!!!!\n");  
  
FILE *f;  
  
f = stdout;  
  
fprintf(f, "Salam Kako 4!!!!\n");
```

## Open a file -> stream

- Read from stream
- Write to stream

```
nasihatkon@kntu:code$ gcc file1.c && ./a.out  
Salam Kako!!!!  
Salam Kako 2!!!!  
Salam Kako 3!!!!  
Salam Kako 4!!!!
```



1926

K. N. Toet University of Technology

# Writing to streams

```
extern struct _IO_FILE *stdin;  
extern struct _IO_FILE *stdout;  
extern struct _IO_FILE *stderr;
```

```
struct _IO_FILE;
```

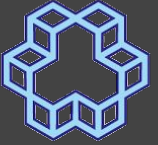
```
typedef struct _IO_FILE FILE;
```

```
printf("Salam Kako!!!!\n");  
  
fprintf(stdout, "Salam Kako 2!!!!\n");  
  
fprintf(stderr, "Salam Kako 3!!!!\n");  
  
FILE *f;  
  
f = stdout;  
  
fprintf(f, "Salam Kako 4!!!!\n");
```

## Open a file -> stream

- Read from stream
- Write to stream

```
nasihatkon@kntu:code$ gcc file1.c && ./a.out > tmp.txt  
Salam Kako 3!!!!  
nasihatkon@kntu:code$ gcc file1.c && ./a.out 2> tmp.txt  
Salam Kako!!!!  
Salam Kako 2!!!!  
Salam Kako 4!!!!
```



1926

K. J. Somaiya Institute of Technology

# Writing to files (File -> stream)

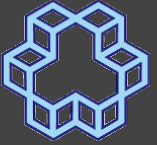
```
extern struct _IO_FILE *stdin;  
extern struct _IO_FILE *stdout;  
extern struct _IO_FILE *stderr;
```

```
struct _IO_FILE;  
typedef struct _IO_FILE FILE;
```

```
FILE *stream;  
  
stream = fopen("salam.txt", "w");  
  
fprintf(stream, "Salam Kako!!!!!!!!!!!!\n");  
  
fclose(stream);
```

## Open a file -> stream

- Read from stream
- Write to stream



1926

K. J. Somaiya Institute of Technology

# Writing to files

```
extern struct _IO_FILE *stdin;  
extern struct _IO_FILE *stdout;  
extern struct _IO_FILE *stderr;
```

```
struct _IO_FILE;
```

```
typedef struct _IO_FILE FILE;
```

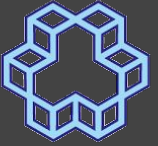
```
FILE *stream;
```

```
stream = fopen("salam.txt", "w");
```

```
fprintf(stream, "Salam Kako!!!!!!!!!!!!\n");
```

```
fclose(stream);
```

```
nasihatkon@kntu:code$ gcc file2.c && ./a.out  
nasihatkon@kntu:code$ cat salam.txt  
Salam Kako!!!!!!!!!!!!
```



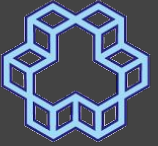
1926

K. N. Toosi University of Technology

# Writing to files

```
FILE *stream = fopen("salam.txt", "w");  
  
fprintf(stream, "Salam!\n");  
  
fprintf(stream, "x=%d\n", -1204);  
  
fprintf(stream, "y=%f, pi=%f\n", 10.5, 3.1415);  
  
fclose(stream);
```





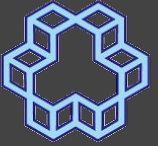
1926

K. J. Somaiya Institute of Technology

# Writing to files

```
FILE *stream = fopen("salam.txt", "w");  
  
fprintf(stream, "Salam!\n");  
  
fprintf(stream, "x=%d\n", -1204);  
  
fprintf(stream, "y=%f, pi=%f\n", 10.5, 3.1415);  
  
fclose(stream);
```

```
nasihatkon@kntu:code$ gcc file3.c && ./a.out  
nasihatkon@kntu:code$ cat salam.txt  
Salam!  
x=-1204  
y=10.500000, pi=3.141500  
nasihatkon@kntu:code$
```



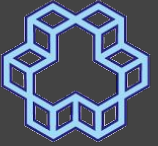
1926

K. N. Toosi University of Technology

# Writing to files

```
FILE *stream;  
  
stream = fopen("nums.txt", "w");  
  
for (int i = 1; i <= 10; i++)  
    fprintf(stream, "%d\n", i);  
  
fclose(stream);
```





1926

K. J. Somaiya Institute of Technology

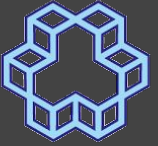
# Writing to files

```
FILE *stream;  
  
stream = fopen("nums.txt", "w");  
  
for (int i = 1; i <= 10; i++)  
    fprintf(stream, "%d\n", i);  
  
fclose(stream);
```

```
nasihatkon@kntu:code$ gcc file4.c && ./a.out  
nasihatkon@kntu:code$ cat nums.txt
```

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

# Writing to files



1926

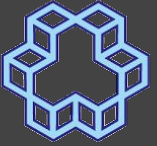
K. J. Somaiya Institute of Technology

```
FILE *stream = fopen("timestable", "w");

for (int i = 1; i <= 10; i++) {
    for (int j = 1; j <= 10; j++)
        fprintf(stream, "%4d", i*j);

    fputc('\n', stream);
}

fclose(stream);
```



1926

K. J. Somaiya Institute of Technology

# Writing to files

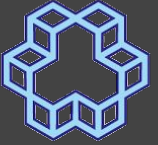
```
FILE *stream = fopen("timestable", "w");

for (int i = 1; i <= 10; i++) {
    for (int j = 1; j <= 10; j++)
        fprintf(stream, "%4d", i*j);

    fputc('\n', stream);
}

fclose(stream);
```

```
nasihatkon@kntu:code$ gcc file5.c && ./a.out
nasihatkon@kntu:code$
nasihatkon@kntu:code$ cat timestable
 1  2  3  4  5  6  7  8  9 10
 2  4  6  8 10 12 14 16 18 20
 3  6  9 12 15 18 21 24 27 30
 4  8 12 16 20 24 28 32 36 40
 5 10 15 20 25 30 35 40 45 50
 6 12 18 24 30 36 42 48 54 60
 7 14 21 28 35 42 49 56 63 70
 8 16 24 32 40 48 56 64 72 80
 9 18 27 36 45 54 63 72 81 90
10 20 30 40 50 60 70 80 90 100
```



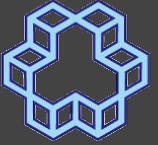
1926

K. J. Somaiya Institute of Technology

# Write to stream functions

standard output	arbitrary stream
<code>printf("format", ...)</code>	<code>fprintf(stream, "format", ...)</code>
<code>putchar(c)</code>	<code>fputc(c, stream)</code> <code>putc(c, stream)</code>
<code>puts("string")</code> (prints trailing '\n', but not '\0')	<code>fputs("string", stream)</code> (without trailing '\n' or '\0')

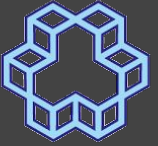
# Appending to files



1926

K. J. Somaiya Institute of Technology

```
int main() {  
  
    FILE *stream = fopen("salam.txt", "w");  
  
    fprintf(stream, "salam!\n");  
  
    fclose(stream);  
  
    return 0;  
}
```



1926

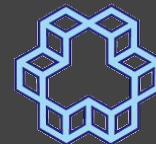
K. J. Somaiya Institute of Technology

# Appending to files

```
int main() {  
  
    FILE *stream = fopen("salam.txt", "w");  
  
    fprintf(stream, "salam!\n");  
  
    fclose(stream);  
  
    return 0;  
}
```

```
nasihatkon@kntu:code$ cat salam.txt  
HELLO  
nasihatkon@kntu:code$ gcc file6.c  
nasihatkon@kntu:code$ ./a.out  
nasihatkon@kntu:code$ cat salam.txt  
salam!  
nasihatkon@kntu:code$ ./a.out  
nasihatkon@kntu:code$ cat salam.txt  
salam!
```

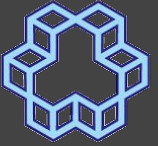
# Appending to files



1926

K. N. Toosi University of Technology

```
int main() {  
  
    FILE *stream = fopen("salam.txt", "a");  
  
    fprintf(stream, "salam!\n");  
  
    fclose(stream);  
  
    return 0;  
}
```



1926

K. J. Somaiya Institute of Technology

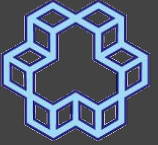
# Appending to files

```
int main() {  
    FILE *stream = fopen("salam.txt", "a");  
    fprintf(stream, "salam!\n");  
    fclose(stream);  
  
    return 0;  
}
```

```
nasihatkon@kntu:code$ cat salam.txt  
salam!  
nasihatkon@kntu:code$ gcc file7.c  
nasihatkon@kntu:code$ ./a.out  
nasihatkon@kntu:code$ cat salam.txt  
salam!  
salam!  
nasihatkon@kntu:code$ ./a.out  
nasihatkon@kntu:code$ cat salam.txt  
salam!  
salam!  
salam!  
nasihatkon@kntu:code$ ./a.out  
nasihatkon@kntu:code$ cat salam.txt  
salam!  
salam!  
salam!  
salam!
```



# Reading a file



1926

K. J. Somaiya Institute of Technology

input1.txt

Salam  
Salaaaaaaaaam!  
Chetori?  
Kako?

```
char buffer[1001];  
  
FILE *stream = fopen("input1.txt", "r");  
  
fscanf(stream, "%s", buffer);  
puts(buffer);  
  
fscanf(stream, "%s", buffer);  
puts(buffer);  
  
fscanf(stream, "%s", buffer);  
puts(buffer);  
  
fscanf(stream, "%s", buffer);  
puts(buffer);  
  
fclose(stream);
```



1926

K. J. Somaiya Institute of Technology

# Reading a file

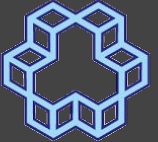
input1.txt

```
Salam  
Salaaaaaaaaam!  
Chetori?  
Kako?
```

```
nasihatkon@kntu:code$ gcc file8.c && ./a.out  
Salam  
Salaaaaaaaaam!  
Chetori?  
Kako?
```

```
char buffer[1001];  
  
FILE *stream = fopen("input1.txt", "r");  
  
fscanf(stream, "%s", buffer);  
puts(buffer);  
  
fscanf(stream, "%s", buffer);  
puts(buffer);  
  
fscanf(stream, "%s", buffer);  
puts(buffer);  
  
fscanf(stream, "%s", buffer);  
puts(buffer);  
  
fclose(stream);
```

# Reading a file



1926

K. J. Somaiya Institute of Technology

input1.txt

Salam  
Salaaaaaaaaam!  
Chetori?  
Kako?

```
#include <stdio.h>

int main() {

    char buffer[1000];

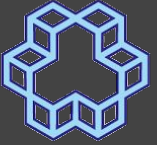
    FILE *stream = fopen("input1.txt", "r");

    for (int i = 0; i < 10; i++) {
        fscanf(stream, "%s", buffer);
        puts(buffer);
    }

    fclose(stream);

    return 0;
}
```

# Reading a file



1926

K. J. Somaiya Institute of Technology

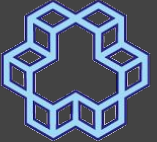
input1.txt

```
Salam  
Salaaaaaaaaam!  
Chetori?  
Kako?
```

```
nasihatkon@kntu:code$ gcc file9.c && ./a.out  
Salam  
Salaaaaaaaaam!  
Chetori?  
Kako?  
Kako?  
Kako?  
Kako?  
Kako?  
Kako?  
Kako?  
Kako?
```

```
#include <stdio.h>  
  
int main() {  
  
    char buffer[1000];  
  
    FILE *stream = fopen("input1.txt", "r");  
  
    for (int i = 0; i < 10; i++) {  
        fscanf(stream, "%s", buffer);  
        puts(buffer);  
    }  
  
    fclose(stream);  
  
    return 0;  
}
```

# Reading a file



1926

K. N. Toosi University of Technology

input1.txt

```
Salam
Salaaaaaaaaaam!
Chetori?
Kako?
```

```
nasihatkon@kntu:code$ gcc file9.c && ./a.out
Salam
Salaaaaaaaaaam!
Chetori?
Kako?
Kako?
Kako?
Kako?
Kako?
Kako?
Kako?
Kako?
```

what's going on?

```
#include <stdio.h>

int main() {

    char buffer[1000];

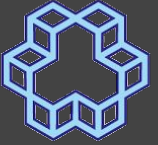
    FILE *stream = fopen("input1.txt", "r");

    for (int i = 0; i < 10; i++) {
        fscanf(stream, "%s", buffer);
        puts(buffer);
    }

    fclose(stream);

    return 0;
}
```

# Reading a file



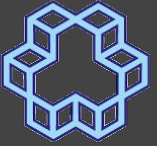
1926

K. J. Somaiya Institute of Technology

`input1.txt`

Salam  
Salaaaaaaaaam!  
Chetori?  
Kako?

```
char buffer[1000];  
FILE *stream = fopen("input1.txt", "r");  
for (int i = 0; i < 10; i++) {  
    int ret = fscanf(stream, "%s", buffer);  
    printf("%d: %s\n", ret, buffer);  
}  
fclose(stream);
```



1926

K. J. Somaiya Institute of Technology

# Reading a file

`input1.txt`

```
Salam  
Salaaaaaaaaam!  
Chetori?  
Kako?
```

```
nasihatkon@kntu:code$ gcc file10.c && ./a.out  
1: Salam  
1: Salaaaaaaaaam!  
1: Chetori?  
1: Kako?  
-1: Kako?  
-1: Kako?  
-1: Kako?  
-1: Kako?  
-1: Kako?  
-1: Kako?
```

```
char buffer[1000];  
  
FILE *stream = fopen("input1.txt", "r");  
  
for (int i = 0; i < 10; i++) {  
  
    int ret = fscanf(stream, "%s", buffer);  
  
    printf("%d: %s\n", ret, buffer);  
}  
  
fclose(stream);
```



1926

K. J. Somaiya Institute of Technology

# Reading a file

input1.txt

```
Salam
Salaaaaaaaaaam!
Chetori?
Kako?
```

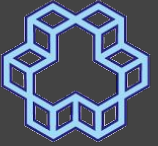
```
nasihatkon@kntu:code$ gcc file10.c && ./a.out
1: Salam
1: Salaaaaaaaaaam!
1: Chetori?
1: Kako?
-1: Kako?
-1: Kako?
-1: Kako?
-1: Kako?
-1: Kako?
-1: Kako?
```

```
char buffer[1000];
FILE *stream = fopen("input1.txt", "r");
for (int i = 0; i < 10; i++) {
    int ret = fscanf(stream, "%s", buffer);
    printf("%d: %s\n", ret, buffer);
}
fclose(stream);
```

stdio.h

```
#ifndef EOF
# define EOF (-1)
#endif
```





1926

K. N. Toosi University of Technology

# Reading a file

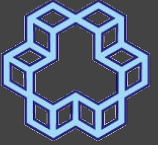
`input1.txt`

```
Salam  
Salaaaaaaaaam!  
Chetori?  
Kako?
```

```
char buffer[1000];  
  
FILE *stream = fopen("input1.txt", "r");  
  
while ( fscanf(stream, "%s", buffer) != EOF ) {  
    puts(buffer);  
}  
  
fclose(stream);
```

```
nasihatkon@kntu:code$ gcc file12.c && ./a.out  
Salam  
Salaaaaaaaaam!  
Chetori?  
Kako?  
nasihatkon@kntu:code$
```

# Reading a file



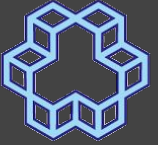
1926

K. J. Somaiya Institute of Technology

`input2.txt`

```
78.4781 53.8523
84.8712 83.4061
39.1451 53.3214
76.6731 46.5567
98.4300 70.6770
23.6938 88.3916
63.9619 69.9066
43.9198 83.0578
34.5258 82.6090
84.4959 34.9013
63.6365 41.0624
 9.8199 30.3438
55.9582 19.7075
51.8794 28.2899
```

# Reading a file



1926

K. N. Toosi University of Technology

input2.txt

```
78.4781 53.8523
84.8712 83.4061
39.1451 53.3214
76.6731 46.5567
98.4300 70.6770
23.6938 88.3916
63.9619 69.9066
43.9198 83.0578
34.5258 82.6090
84.4959 34.9013
63.6365 41.0624
 9.8199 30.3438
55.9582 19.7075
51.8794 28.2899
```

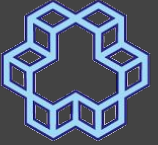
```
double data[100][2];
int n = 0;

FILE *stream = fopen("input2.txt", "r");

while ( fscanf(stream, "%lf %lf", &data[n][0], &data[n][1]) != EOF )
    n++;

fclose(stream);
```

# Reading a file



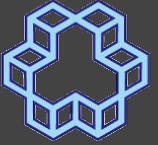
1926

K. N. Toosi University of Technology

`input3.txt`

```
96324532 Amin Parchami 18.5
96234534 Behnam Beigi 17.9
96838459 Parham Parviz 18.9
96838222 Mahdi Forozan 19.99
```

# Reading a file



1926

K. N. Toosi University of Technology

input3.txt

```
96324532 Amin Parchami 18.5
96234534 Behnam Beigi 17.9
96838459 Parham Parviz 18.9
96838222 Mahdi Forozan 19.99
```

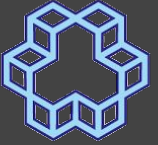
```
int id;
char fName[100];
char lName[200];
double avg;

int n = 0;
FILE *stream = fopen("input3.txt", "r");

while ( fscanf(stream, "%d %s %s %lf", &id, fName, lName, &avg) != EOF ) {
    printf("ID: %d, Name: %-7s %-8s, average: %2.2f\n", id, fName, lName, avg );
}

fclose(stream);
```

# Reading a file



1926

K. N. Toosi University of Technology

input3.txt

```
96324532 Amin Parchami 18.5
96234534 Behnam Beigi 17.9
96838459 Parham Parviz 18.9
96838222 Mahdi Forozan 19.99
```

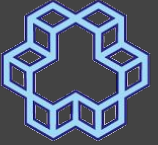
```
nasihatkon@kntu:code$ gcc file14.c && ./a.out
ID: 96324532, Name: Amin Parchami, average: 18.50
ID: 96234534, Name: Behnam Beigi , average: 17.90
ID: 96838459, Name: Parham Parviz , average: 18.90
ID: 96838222, Name: Mahdi Forozan , average: 19.99
```

```
int id;
char fName[100];
char lName[200];
double avg;

int n = 0;
FILE *stream = fopen("input3.txt", "r");

while ( fscanf(stream, "%d %s %s %lf", &id, fName, lName, &avg) != EOF ) {
    printf("ID: %d, Name: %-7s %-8s, average: %2.2f\n", id, fName, lName, avg );
}

fclose(stream);
```

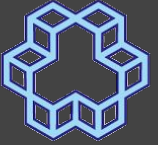


1926

K. J. Somaiya Institute of Technology

# Reading streams functions

standard input	arbitrary stream
<code>scanf("format", ...)</code>	<code>fscanf(stream, "format", ...)</code>
<code>c = getchar()</code>	<code>c = fgetc(stream)</code> <code>c = getc(stream)</code>
<code>gets(s)</code>	<code>fgets(s, size, stream)</code>



1926

K. J. Somaiya Institute of Technology

# Write your own cat

```
int c;
FILE *stream;
char fname[100];

printf("File name: ");
scanf("%99s", fname);

stream = fopen(fname, "r");

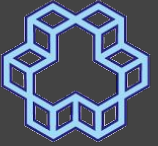
if (stream == NULL) {
    fprintf(stderr, "Cannot open file!\n");
    return 1;
}

while ( (c = fgetc(stream) ) != EOF ) {
    putchar(c);
}

fclose(stream);

return 0;
```





1926

K. J. Somaiya Institute of Technology

# Write your own cat

```
int c;
FILE *stream;
char fname[100];

printf("File name: ");
scanf("%99s", fname);

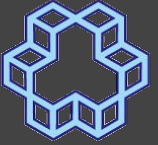
stream = fopen(fname, "r");

if (stream == NULL) {
    fprintf(stderr, "Cannot open file!\n");
    return 1;
}

while ( (c = fgetc(stream) ) != EOF ) {
    putchar(c);
}

fclose(stream);

return 0;
```



1926

K. J. Somaiya Institute of Technology

# Write your own cat

```
int c;
FILE *stream;
char fname[100];

printf("File name: ");
scanf("%99s", fname);

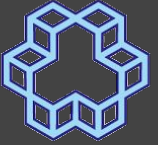
stream = fopen(fname, "r");

if (stream == NULL) {
    fprintf(stderr, "Cannot open file!\n");
    return 1;
}

while ( (c = fgetc(stream) ) != EOF ) {
    putchar(c);
}

fclose(stream);

return 0;
```



1926

K. J. Somaiya Institute of Technology

# Write your own cat

```
int c;
FILE *stream;
char fname[100];

printf("File name: ");
scanf("%99s", fname);

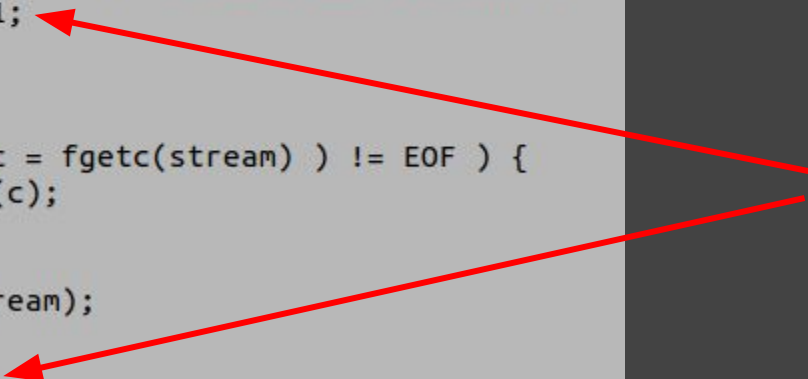
stream = fopen(fname, "r");

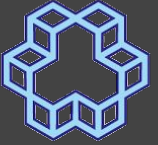
if (stream == NULL) {
    fprintf(stderr, "Cannot open file!\n");
    return 1;
}

while ( (c = fgetc(stream) ) != EOF ) {
    putchar(c);
}

fclose(stream);

return 0;
```





1926

K. N. Toosi University of Technology

# Write your own cat

```
int c;
FILE *stream;
char fname[100];

printf("File name: ");
scanf("%99s", fname);

stream = fopen(fname, "r");

if (stream == NULL) {
    fprintf(stderr, "Cannot open file!\n");
    return 1;
}

while ( (c = fgetc(stream)) != EOF ) {
    putchar(c);
}

fclose(stream);

return 0;
```

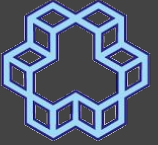
why *int* instead of *char*?



```
int fgetc(FILE *stream);
```

## RETURN VALUE

`fgetc()`, `getc()` and `getchar()` return the character read as an unsigned char cast to an int or `EOF` on end of file or error.



1926

K. J. Somaiya Institute of Technology

# Write your own cat

```
int c;
FILE *stream;
char fname[100];

printf("File name: ");
scanf("%99s", fname);

stream = fopen(fname, "r");

if (stream == NULL) {
    fprintf(stderr, "Cannot open file!\n");
    return 1;
}

while ( (c = fgetc(stream) ) != EOF ) {
    putchar(c);
}

fclose(stream);

return 0;
```

```
nasihatkon@kntu:code$ gcc file15.c -o cat && ./cat
File name: input1.txt
Salam
Salaaaaaaaaam!
Chetori?
Kako?
```



1926

K. N. Toosi University of Technology

# Write your own cat

```
int c;
FILE *stream;
char fname[100];

printf("File name: ");
scanf("%99s", fname);

stream = fopen(fname, "r");

if (stream == NULL) {
    fprintf(stderr, "Cannot open file!\n");
    return 1;
}

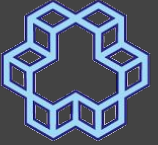
while ( (c = fgetc(stream) ) != EOF ) {
    putchar(c);
}

fclose(stream);

return 0;
```

```
nasihatkon@kntu:code$ gcc file15.c -o cat && ./cat
File name: input1.txt
Salam
Salaaaaaaaaaam!
Chetori?
Kako?
```

```
nasihatkon@kntu:code$ ./cat
File name: input3.txt
96324532   Amin   Parchami   18.5
96234534   Behnam Beigi     17.9
96838459   Parham Parviz    18.9
96838222   Mahdi  Forozan    19.99
```



1926

K. N. Toosi University of Technology

# Using feof

```
int c;
FILE *stream;
char fname[100];

printf("File name: ");
scanf("%99s", fname);

stream = fopen(fname, "r");

if (stream == NULL) {
    fprintf(stderr, "Cannot open file!\n");
    return 1;
}

while ( (c = fgetc(stream) ) != EOF ) {
    putchar(c);
}

fclose(stream);

return 0;
```

```
int c;
FILE *stream;
char fname[100];

printf("File name: ");
scanf("%99s", fname);

stream = fopen(fname, "r");

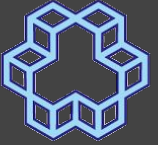
if (stream == NULL) {
    fprintf(stderr, "Cannot open file!\n");
    exit(1);
}

while ( ! feof(stream) ) {
    putchar(fgetc(stream));
}

fclose(stream);

return 0;
```

# EOF in stdin



1926

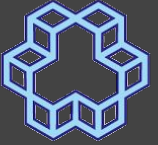
K. J. Somaiya Institute of Technology

```
int c;  
  
while ( (c = fgetc(stdin)) != EOF ) {  
    putchar(c);  
}
```

**linux: Ctrl+d**

**windows: Ctrl+z**





1926

K. N. Toosi University of Technology

# rewinding a file stream

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

int main() {
    int c;
    FILE *stream;

    stream = fopen("input1.txt", "r");

    while ( (c = fgetc(stream) ) != EOF ) {
        putchar(c);
    }

    rewind(stream);

    while ( (c = fgetc(stream) ) != EOF ) {
        putchar(toupper(c));
    }

    fclose(stream);

    return 0;
}
```



1926

K. J. Somaiya Institute of Technology

# rewinding a file stream

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

int main() {
    int c;
    FILE *stream;

    stream = fopen("input1.txt", "r");

    while ( (c = fgetc(stream) ) != EOF ) {
        putchar(c);
    }

    rewind(stream);

    while ( (c = fgetc(stream) ) != EOF ) {
        putchar(toupper(c));
    }

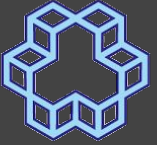
    fclose(stream);

    return 0;
}
```

```
nasihatkon@kntu:code$ gcc file18.c && ./a.out
Salam
Salaaaaaaaaaam!
Chetori?
Kako?

SALAM
SALAAAAAAAAAAM!
CHETORI?
KAKO?
```

# using fseek



1926

K. N. Toosi University of Technology

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

int main() {
    int c;
    FILE *stream;

    stream = fopen("input1.txt", "r");

    while ( (c = fgetc(stream) ) != EOF ) {
        putchar(c);
    }

    rewind(stream);

    while ( (c = fgetc(stream) ) != EOF ) {
        putchar(toupper(c));
    }

    fclose(stream);

    return 0;
}
```

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

int main() {
    int c;
    FILE *stream;

    stream = fopen("input1.txt", "r");

    while ( (c = fgetc(stream) ) != EOF ) {
        putchar(c);
    }

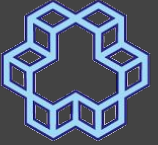
    fseek(stream,0, SEEK_SET);

    while ( (c = fgetc(stream) ) != EOF ) {
        putchar(toupper(c));
    }

    fclose(stream);

    return 0;
}
```

# using fseek



1926

K. J. Somaiya Institute of Technology

```
int c;
FILE *stream = fopen("input1.txt", "r");

while ( (c = fgetc(stream) ) != EOF )
    putchar(c);

fseek(stream,10, SEEK_SET);

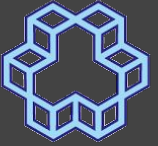
while ( (c = fgetc(stream) ) != EOF )
    putchar(c);

fclose(stream);
```

```
nasihatkon@kntu:code$ gcc file20.c && ./a.out
Salam
Salaaaaaaaaam!
Chetori?
Kako?

aaaaaaaaam!
Chetori?
Kako?
```

# using fseek



1926

K. J. Somaiya Institute of Technology

```
int c;
FILE *stream = fopen("input1.txt", "r");

while ( (c = fgetc(stream) ) != EOF )
    putchar(c);

fseek(stream, -20, SEEK_END);

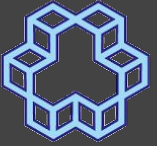
while ( (c = fgetc(stream) ) != EOF )
    putchar(c);

fclose(stream);
```

```
nasihatkon@kntu:code$ gcc file21.c && ./a.out
Salam
Salaaaaaaaaam!
Chetori?
Kako?

m!
Chetori?
Kako?
```

# using fseek



1926

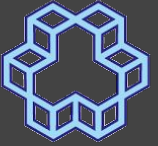
K. N. Toosi University of Technology

<code>fseek(stream, 10, SEEK_SET);</code>	sets position indicator to 10 (from the beginning of the file)
<code>fseek(stream, -20, SEEK_END);</code>	sets position indicator to 20 bytes to the end of the file (end of file - 20)
<code>fseek(stream, 30, SEEK_CUR);</code>	increase position indicator by 30 (current position + 30)
<code>fseek(stream, -5, SEEK_CUR);</code>	decrease position indicator by 5 (current position - 5)

`stdio.h`

```
#define SEEK_SET      0
#define SEEK_CUR     1
#define SEEK_END     2
```

# using ftell



1926

K. J. Somaiya Institute of Technology

```
int c;
FILE *fp = fopen("input1.txt", "r");

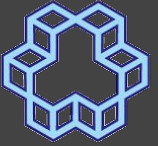
for (int i = 0; i < 20; i++)
    c = fgetc(fp);

printf("Position=%ld\n", ftell(fp));

fseek(fp, -7, SEEK_CUR);

printf("Position=%ld\n", ftell(fp));

fclose(fp);
```



1926

K. J. Somaiya Institute of Technology

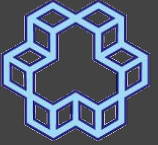
# using ftell

```
int c;  
FILE *fp = fopen("input1.txt", "r");  
  
for (int i = 0; i < 20; i++)  
    c = fgetc(fp);  
  
printf("Position=%ld\n", ftell(fp));  
  
fseek(fp, -7, SEEK_CUR);  
  
printf("Position=%ld\n", ftell(fp));  
  
fclose(fp);
```

```
nasihatkon@kntu:code$ gcc file22.c && ./a.out  
Position=20  
Position=13
```



# fopen modes



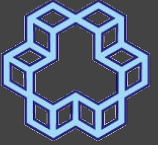
1926

K. J. Somaiya Institute of Technology

<b>r</b>	<b>Opens file for reading. File must exist.</b>
<b>w</b>	<b>Creates an empty file for writing.</b>
<b>a</b>	<b>Opens file for appending. File created if not exist.</b>
<b>r+</b>	<b>Opens files for reading and writing. File must exist.</b>
<b>w+</b>	<b>Creates an empty file for reading and writing.</b>
<b>a+</b>	<b>Opens file for reading and appending.</b>

[https://www.tutorialspoint.com/c\\_standard\\_library/c\\_function\\_fopen.htm](https://www.tutorialspoint.com/c_standard_library/c_function_fopen.htm)

# Updating records



1926

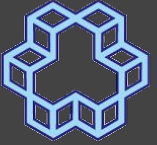
K. N. Toosi University of Technology

`input.txt`

```
96324532 Amin Parchami 18.5
96234534 Behnam Beigi 17.9
96838459 Parham Parviz 18.9
96838222 Mahdi Forozan 19.99
```

How to

- go to n-th record?
- change a record?



1926

K. N. Toosi University of Technology

# Fixed-sized records

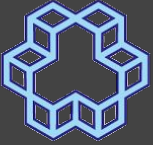
ASCII Hex Symbol			ASCII Hex Symbol		
32	20	(space)	48	30	0
33	21	!	49	31	1
34	22	"	50	32	2
35	23	#	51	33	3
36	24	\$	52	34	4
37	25	%	53	35	5
38	26	&	54	36	6
39	27	'	55	37	7
40	28	(	56	38	8
41	29	)	57	39	9
42	2A	*	58	3A	:
43	2B	+	59	3B	;
44	2C	,	60	3C	<
45	2D	-	61	3D	=
46	2E	.	62	3E	>
47	2F	/	63	3F	?

12

- {'1', '2'} = {49, 50} (2 bytes)
- 00001100 (1 byte)

228

- {'2', '2', '8'} = {50, 50, 56} (3 bytes)
- 11100100 (1 byte)



1926

K. N. Toosi University of Technology

# Fixed-sized records

ASCII Hex Symbol			ASCII Hex Symbol		
32	20	(space)	48	30	0
33	21	!	49	31	1
34	22	"	50	32	2
35	23	#	51	33	3
36	24	\$	52	34	4
37	25	%	53	35	5
38	26	&	54	36	6
39	27	'	55	37	7
40	28	(	56	38	8
41	29	)	57	39	9
42	2A	*	58	3A	:
43	2B	+	59	3B	;
44	2C	,	60	3C	<
45	2D	-	61	3D	=
46	2E	.	62	3E	>
47	2F	/	63	3F	?

12

- {'1', '2'} = {49, 50} (2 bytes)
- 00001100 (1 byte)

228

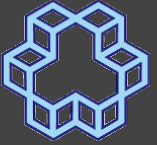
- {'2', '2', '8'} = {50, 50, 56} (3 bytes)
- 11100100 (1 byte)

```
fprintf(stream, "%d", 12);
```



```
fprintf(stream, "%d", 228);
```





1926

K. N. Toosi University of Technology

# Fixed-sized records

ASCII Hex Symbol			ASCII Hex Symbol		
32	20	(space)	48	30	0
33	21	!	49	31	1
34	22	"	50	32	2
35	23	#	51	33	3
36	24	\$	52	34	4
37	25	%	53	35	5
38	26	&	54	36	6
39	27	'	55	37	7
40	28	(	56	38	8
41	29	)	57	39	9
42	2A	*	58	3A	:
43	2B	+	59	3B	;
44	2C	,	60	3C	<
45	2D	-	61	3D	=
46	2E	.	62	3E	>
47	2F	/	63	3F	?

12

- {'1', '2'} = {49, 50} (2 bytes)
- 00001100 (1 byte)

228

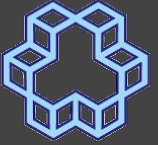
- {'2', '2', '8'} = {50, 50, 56} (3 bytes)
- 11100100 (1 byte)

```
fprintf(stream, "%d", 12);
```



```
fprintf(stream, "%d", 228);
```





1926

K. J. Somaiya Institute of Technology

# Using fwrite: directly writing bytes to file

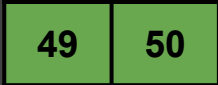
12

- {'1', '2'} = {49, 50} (2 bytes)
- 00001100 (1 byte)

228

- {'2', '2', '8'} = {50,50,56} (3 bytes)
- 11100100 (1 byte)

```
fprintf(stream, "%d", 12);
```



```
fprintf(stream, "%d", 228);
```

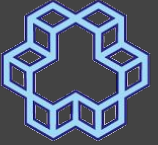


```
char c = 12; fwrite(&c, 1, 1, stream);
```



```
char c = 228; fwrite(&c, 1, 1, stream);
```





1926

K. N. Toosi University of Technology

# Using fwrite: directly writing bytes to file

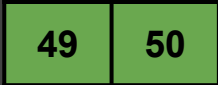
12

- {'1', '2'} = {49, 50} (2 bytes)
- 00001100 (1 byte)

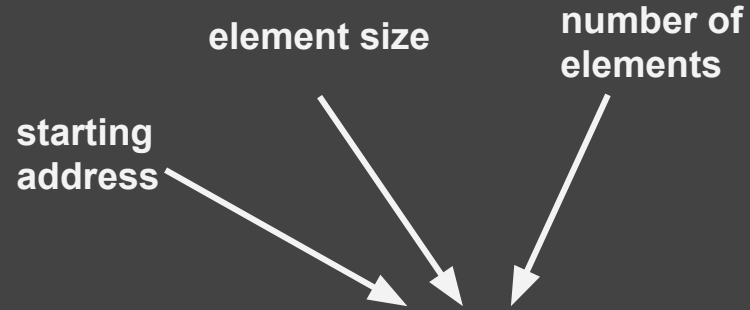
228

- {'2', '2', '8'} = {50,50,56} (3 bytes)
- 11100100 (1 byte)

```
fprintf(stream, "%d", 12);
```



```
fprintf(stream, "%d", 228);
```



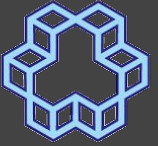
```
char c = 12; fwrite(&c, 1, 1, stream);
```



```
char c = 228; fwrite(&c, 1, 1, stream);
```



```
size_t fwrite(const void *ptr, size_t size, size_t nmem, FILE *stream);
```



1926

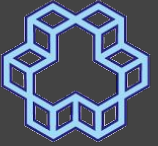
K. J. Somaiya Institute of Technology

# Writing an array to file

```
int array[] = {10,20,30,40,50,60,70};  
int n = sizeof(array)/sizeof(int);  
  
FILE *f = fopen("arrayfile", "w");  
  
fwrite(array, sizeof(int), n, f);  
  
fclose(f);
```

```
fwrite(array, sizeof(int), n, f);
```





1926

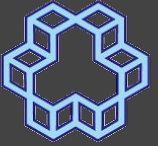
K. J. Somaiya Institute of Technology

# Writing an array to file

```
int array[] = {10,20,30,40,50,60,70};  
int n = sizeof(array)/sizeof(int);  
  
FILE *f = fopen("arrayfile", "w");  
  
fwrite(array, sizeof(int), n, f);  
  
fclose(f);
```

```
fwrite(array, sizeof(int), n, f);
```

```
fwrite(array, sizeof(int)*n, 1, f);
```



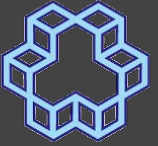
1926

K. J. Somaiya Institute of Technology

# Writing an array to file

```
int array[] = {10,20,30,40,50,60,70};  
int n = sizeof(array)/sizeof(int);  
  
FILE *f = fopen("arrayfile", "w");  
  
fwrite(array, sizeof(int), n, f);  
  
fclose(f);
```

```
int buffer[n];  
  
f = fopen("arrayfile", "r");  
  
fread(buffer, sizeof(int), n, f);  
  
fclose(f);  
  
printArray(buffer, n);
```



1926

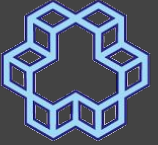
K. J. Somaiya Institute of Technology

# Writing an array to file

```
int array[] = {10,20,30,40,50,60,70};  
int n = sizeof(array)/sizeof(int);  
  
FILE *f = fopen("arrayfile", "w");  
  
fwrite(array, sizeof(int), n, f);  
  
fclose(f);
```

```
int buffer[n];  
  
f = fopen("arrayfile", "r");  
  
fread(buffer, sizeof(int), n, f);  
  
fclose(f);  
  
printArray(buffer, n);
```

# Remember: struct Student

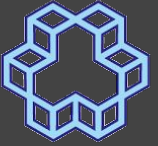


1926

K. J. Somaiya Institute of Technology

```
struct Date {
    int year;
    int month;
    int day;
};

struct Student {
    char firstName[20];
    char lastName[20];
    int id;
    struct Date DoB; // date of birth
    char gender;
};
```

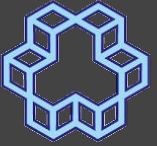


1926

K. N. Toosi University of Technology

# Writing records to a file

```
struct Student s0 = {"Amin", "Parchami", 9664321, {1378,6,7}, 'M'};  
struct Student s1 = {"Behnam", "Beigi", 9634253, {1340, 12, 25}, 'M'};  
struct Student s2 = {"Parham", "Parviz", 9683746, {1390, 12, 30}, 'M'};  
struct Student s3 = {"Mahdi", "Forozan", 9609347, {1380, 10, 5}, 'M'};  
struct Student s4 = {"Parvin", "Etesami", 9600000, {1280, 1, 4}, 'F'};  
  
FILE *f = fopen("records", "w");  
  
fwrite(&s0, sizeof(struct Student), 1, f);  
fwrite(&s1, sizeof(struct Student), 1, f);  
fwrite(&s2, sizeof(struct Student), 1, f);  
fwrite(&s3, sizeof(s3), 1, f);  
  
fclose(f);
```



1926

K. J. Somaiya University of Technology

# Read/Write n-th record in a file

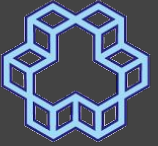
```
struct Student read_record(int rec_no, FILE *f) {
    fseek(f, rec_no*sizeof(struct Student), SEEK_SET);

    struct Student s;
    fread(&s, sizeof(struct Student), 1, f);

    return s;
}

void write_record(int rec_no, struct Student s, FILE *f) {
    fseek(f, rec_no*sizeof(struct Student), SEEK_SET);

    fwrite(&s, sizeof(struct Student), 1, f);
}
```



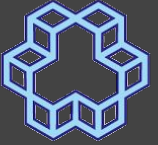
1926

K. J. Somaiya Institute of Technology

## Read/Write n-th record in a file

```
f = fopen("records", "r+");  
struct Student s;  
  
s = read_record(2, f);  
printStudent(&s);  
  
write_record(2, s4, f);  
s = read_record(2, f);  
printStudent(&s);  
  
fclose(f);
```





1926

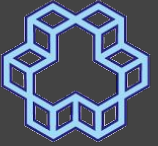
K. N. Toosi University of Technology

# Read/Write n-th record in a file

```
f = fopen("records", "r+");  
struct Student s;  
  
s = read_record(2, f);  
printStudent(&s);  
  
write_record(2, s4, f);  
s = read_record(2, f);  
printStudent(&s);  
  
fclose(f);
```

```
nasihatkon@kntu:code$ gcc file26.c && ./a.out  
First name: Parham  
Last name: Parviz  
ID: 9683746  
DoB: 1390/12/30  
Gender: M  
  
First name: Parvin  
Last name: Etesami  
ID: 9600000  
DoB: 1280/01/04  
Gender: F
```



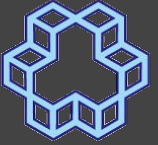


1926

K. J. Somaiya Institute of Technology

# Changing student's name of n-th record

```
void set_name(int rec_no, char *firstName, char* lastName, FILE *f) {  
    struct Student s;  
  
    fseek(f, rec_no*sizeof(struct Student), SEEK_SET);  
    fread(&s, sizeof(struct Student), 1, f);  
  
    strncpy(s.firstName, firstName, 20);  
    strncpy(s.lastName, lastName, 20);  
  
    fseek(f, rec_no*sizeof(struct Student), SEEK_SET);  
    fwrite(&s, sizeof(struct Student), 1, f);  
}
```



1926

K. N. Toosi University of Technology

# Changing student's name of n-th record

```
f = fopen("records", "r+");
struct Student s;

s = read_record(1, f);
printStudent(&s);

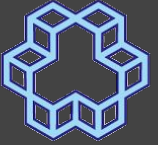
set_name(1, "Behrooz", "Nasihatkon", f);
s = read_record(1, f);
printStudent(&s);

fclose(f);
```

```
nasihatkon@kntu:code$ gcc file27.c && ./a.out
First name: Behnam
Last name: Beigi
ID: 9634253
DoB: 1340/12/25
Gender: M

First name: Behrooz
Last name: Nasihatkon
ID: 9634253
DoB: 1340/12/25
Gender: M
```

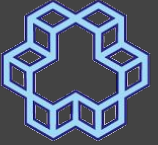
# Record Key



1926

K. J. Somaiya Institute of Technology

Look at example from the book!



1926

K. N. Tceet University of Technology