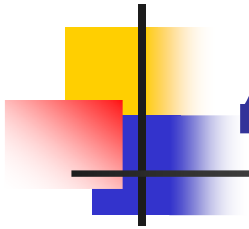


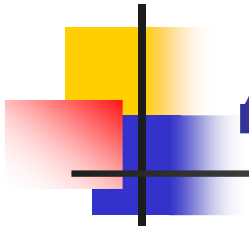
Датотеке – стандардни улаз / излаз

- Датотека је именовани низ знакова (бајтова)
- У програмском језику C датотека је везана за улаз и излаз података – функције стандардне библиотеке
- <stdio.h>
- Најједноставније функције
  - int getchar(void)
  - int putchar(int)
- Конвенција преусмерења, пример
  - prog <indat >outdat 2>greska
- Форматирани излаз / улаз
  - int printf(char *format, *arg*1, *arg*2, ...)
  - int sprintf(char *string, char *format, *arg*1, *arg*2, ...)
  - int scanf(char *format, *arg*1, *arg*2, ...)
  - int sscanf(char *string, char *format, *arg*1, *arg*2, ...)



Датотеке - приступ датотекама

- FILE *fopen(const char *ime, const char *nacin);
- nacin: “r”, “w”, “a”
- FILE *fp;
- fp = fopen(ime, nacin);
- текстуална датотека
  - крај реда -> маркер краја реда (<lf>, <cr><lf>)
  - при читању – 0-карактер
- бинарна датотека: nacin “rb”, “wb”, “ab”
  - не уписује се маркер краја реда, осим експлицитно
  - при читању – чита се дословни садржај
- nacin
  - “r+”, “rb+”, “r+b”- отварање датотеке за ажурирање тј. читање и упис
  - “w+”, “wb+”, “w+b” – креирање датотеке за ажурирање
  - “a+”, “ab+”, “a+b” – отварање или креирање датотеке за ажурирање – упис на крај



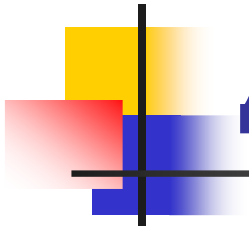
Датотеке – читање и упис

- int fgetc(FILE *fp)
- int fputc(int c, FILE *fp)
- int getc(FILE *fp) (као fgetc, може и као макро)
- int putc(int c, FILE *fp) (као fputc, може и као макро)

- char *fgets(char *s, int maxduzina, FILE *fp),
- int fputs(const char *s, FILE *fp)

- При стартовању програма ОС отвара три стандардне датотеке: **stdin, stdout, stderr**
- #define getchar() getc(stdin)
- #define putchar(c) putc((c), stdout)

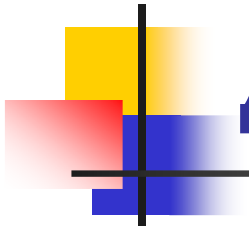
- int fscanf(FILE *fp, char *format, *arg*1, *arg*2, . . .)
- int fprintf(FILE *fp, char *format, *arg*1, *arg*2, . . .)



Датотеке - пример

- Функција која копира садржај датотеке са показивачем *ulazp* у датотеку са показивачем *izlazp* има следећу дефиницију:

```
/*filecopy: kopira datoteku ulazp u datoteku izlazp */
void filecopy(FILE *ulazp, FILE *izlazp)
{ int c;
      while ((c=getc(ulazp)) != EOF)
      putc(c, izlazp);
}
```



Датотеке - пример

- Програм који позива претходну функцију:

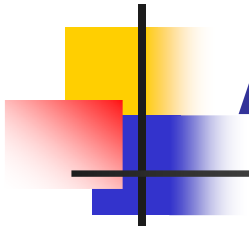
```
#include <stdio.h>
void filecopy(FILE *, FILE *);
/*kopiranje: kopira datoteku a.c u datoteku b.c */
int main()
{      FILE *ulazp, *izlazp;
      if ((ulazp=fopen(”a.c”, ”r”))==NULL) { printf(”ne moze da se otvori datoteka a.c \n”);
                                              return 1;}
      else {if ((izlazp=fopen(”b.c”, ”w”))==NULL) { printf(”ne moze da se otvori datoteka b.c
      \n”);
                                              return 1;}
                else {filecopy(ulazp, izlazp);
                              fclose(izlazp);
                              izlazp=fopen(”b.c”, ”r”);
                              filecopy(izlazp, stdout);
                              fclose(izlazp);
                }
                fclose(ulazp);
              }
}
```

Датотеке – алтернативно преусмеравање

- funkcija freopen()
- <stdio.h>
- FILE *freopen(const char *filename, const char *mode, FILE *stream)

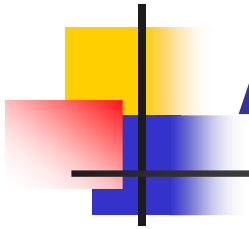
- Пример

```
#include <stdio.h>
int main()
{ int i;
        FILE *ulazp, *izlazp;
        ulazp=freopen(”ulaz.txt”, ”r”, stdin);
        izlazp=freopen(”izlaz.txt”, ”w”, stdout);
        scanf(”%d”, &i);
        printf(”%d \n”, i);
        return 0;
}
```



Аргументи командне линије

- Функција ***main*** до сада је дефинисана без аргумената – ***main()***, а командна линија којом се позива извршавање програма је садржала само један "аргумент" – назив извршног програма.
- У општем случају, функција ***main*** може да има аргументе, а њихове вредности су у тесној вези са бројем и вредностима аргумената командне линије којом се позива извршавање одговарајућег програма
- ***main*** се дефинише са два аргумента : argc (ARGument Count) – број аргумената у командној линији, и argv (ARGument Vector) – вредности аргумената командне линије – то је број стрингова у позиву извршавања програма
- int main(int argc, char *argv[]))



Аргументи командне линије

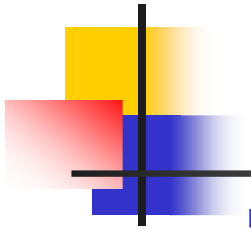
- int main(int argc, char *argv[ ]))
  - argc добија вредност једнаку броју аргумената (рачунајући и само име датотеке извршног програма – то је argv[0]), argv[1] је једнак првом опционом аргументу (који се наводи иза имена извршне датотеке), argv[2] је једнак другом опционом аргументу, итд.
- int main()
  - argc је 1, argv[0] је име програма, argv[1] је NULL

Аргументи командне линије - пример

```
#include <stdio.h>
/*kopiranje: kopira ulaznu datoteku u izlaznu datoteku */
main(int argc, char *argv[ ])
{
      FILE *ulazp, *izlazp;
      void filecopy(FILE *, FILE *);
      if ((ulazp=fopen(*++argv, ”r”))==NULL) {
      printf(”ne moze da se otvori datoteka %s \n”, *argv);
      return 1;
      } else
      {if ((izlazp=fopen(*++argv, ”w”))==NULL) {
                printf(”ne moze da se otvori datoteka %s \n”, *argv);
                return 1;
       }else {
                filecopy(ulazp, izlazp);
                fclose(izlazp);
                izlazp=fopen(*argv, ”r”);
                filecopy(izlazp, stdout);
                fclose(izlazp);
       }
      fclose(ulazp);
      }
}
```

datkopi a.c b.c

Функције директног приступа датотекама: fseek



- int fseek(FILE *stream, long offset, int origin)

- Вредност *origin* може бити SEEK_SET (почетак датотеке), SEEK_CUR (текућа позиција у датотеци) или SEEK_END (крај датотеке). Ове вредности су константе дефинисане у <stdio.h>.

- Функција fseek враћа вредност 0 ако је позиционирање успешно, а вредност различиту од 0 у случају грешке

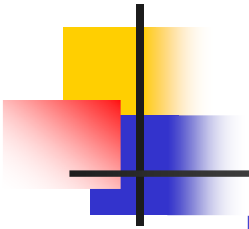
Функције директног приступа датотекама: fseek

- Пример:

```
#include <stdio.h>
int main()
{
        FILE *fp;
        fp=fopen(”izlaz.txt”, ”w”);
        fputs(”This is an apple.”, fp);
        fseek(fp, 9, SEEK_SET);
        fputs(” sam”, fp);
        fclose(fp);
        return 0;
}
```

- ”This is an apple.” → “This is a sample.”

Функције директног приступа датотекама: ftell



- long int ftell(FILE *stream)
- <stdio.h>
- Враћа текућу вредност индикатора позиције у датотеци којој је придружен показивач *stream*.
- -1L у случају грешке

Функције директног приступа датотекама: ftell

- **Пример**

```
#include <stdio.h>
int main()
{
        FILE *fp;
        long size;
        fp=fopen(”ulaz.txt”, ”r”);
        if(fp==NULL){
                printf(”greska pri otvaranju datoteke”);
                return 1;
        }else
        {
                fseek(fp, 0, SEEK_END);
                size = ftell(fp);
                fclose(fp);
                printf(”Velicina datoteke ulaz.txt: %ld bajtova \n”, size);
                return 0;
        }
}
```