





## Vstup/výstup po znacích

{ IVT IV. }  
1

- `int fgetc(FILE* soubor);`
  - Analogie funkce `getchar` (pro standardizované načítání znaků)
- `int fputc(int znak, FILE* soubor);`
  - Analogie funkce `putchar`
  - Pozor! Parametr `soubor` je až na 2. místě.
- `int ungetc (int znak, FILE* soubor);`
  - Vrací jeden znak zpět do řádkového bufferu.
  - Jen jeden znak!
  - Neexistuje varianta pro formátované čtení!

IVT - teorie, kvarta

27/38

## Vstup/výstup po znacích

{ IVT IV. }  
1

```
// Kopíruje obsah vstupního souboru do výstupního.
// vstup - soubor otevřený pro čtení
// výstup - soubor otevřený pro zápis
void kopiePoZnacich(FILE* vstup, FILE* vystup) {
    int znak;
    while ( (znak = fgetc(vstup)) != EOF) {
        fputc(znak, vystup);
    }
}
```

IVT - teorie, kvarta

28/38

## Vstup/výstup po řádcích

{ IVT IV. }  
1

- `char *fgets(char *radek, int n, FILE *soubor);`
  - Čte ze souboru řádek, maximálně ale n-1 znaků.
  - Pole `radek` musí mít délku alespoň n znaků.
  - Čte a ukládá i znak `'\n'`.
  - Vrací NULL při chybě.
- `int fputs(const char *s, FILE *stream);`
  - Vytiskne do souboru zadaný textový řetězec a odřádkuje.
  - Vrací EOF při chybě.

IVT - teorie, kvarta

29/38

## Formátovaný vstup/výstup

{ IVT IV. }  
1

```
// Kopíruje souborů s textově zapsanými čísly.
// Ve výstupu se objeví každé číslo na novém řádku.
// vstup - soubor otevřený pro čtení
// výstup - soubor otevřený pro zápis
void kopiePoCislech(FILE* vstup, FILE* vystup) {
    float cislo;
    while (fscanf(vstup, "%f", &cislo) == 1) {
        fprintf(vystup, "%f\n", cislo);
    }
}
```

IVT - teorie, kvarta

31/38