

# Podstawy programowania komputerów

## Wykład 9: „Operacje na plikach w ANSI C”



# Operacje plikowe(nie obiektowo)

Plik - dokument lub inny zbiór danych zapisany na dysku, oznaczony unikatową nazwą. Plik jest ciągiem bajtów

Plik w C:

- plik dyskowy (**disk file**)
- urządzenie peryferyjne typu **terminal**
- drukarka
- plik na taśmie magnetycznej (**tape drive file**)



# Strumień

- Strumień (**stream**) w C: przepływ danych z programu do pliku lub odwrotnie
- Strumień nie jest związany z konkretnym urządzeniem (**device-independend**)
- Aby przeprowadzić operację Wejścia/Wyjścia, należy skojarzyć plik ze strumieniem
- Dwa formaty strumieni: **text or binary stream**



# Buforowanie Wejścia/Wyjścia

- Bufor (**buffer**) - obszar pamięci służący do tymczasowego przechowywania danych przed przesłaniem ich do miejsca przeznaczenia
- Posługując się buforami system operacyjny zmniejsza ilość odwołań do fizycznych urządzeń Wejścia/Wyjścia
- Strumienie We/Wy są buforowane w sposób domyślny
- Fizyczny zapis do pliku - **buffer flush**



# Wskaźniki typu FILE

- Struktura FILE służy do zarządzania plikami i jest zdefiniowana w pliku **stdio.h**
- Wskaźnik typu FILE - **file pointer**, służy do odwoływania się do konkretnego pliku dyskowego.
- W obrębie struktury FILE można znaleźć: rozmiar pliku, znacznik pozycji w pliku , adres bufora danych itp.



# Otwieranie pliku

- `# include <stdio.h>`

```
FILE *fopen(const char *filename, const  
char *mode);
```

- Jeśli operacja otwarcia pliku nie powiedzie się, funkcja zwraca wskaźnik pusty (**null pointer**).

- Tryby otwarcia pliku:

r (read)	r+	rb	rb+	r+b
w (write)	w+	wb	wb+	w+b
a (append)	a+	ab	ab+	a+b

tryb **w** tworzy plik



# Otwieranie/zamykanie pliku

```
FILE *fptr;  
if ((fptr = fopen („TEST.TXT”, „r”))  
== NULL)  
{  
printf („Nie moze otworzyc pliku  
TEST.TXT.\n”);  
exit (1)  
}
```

- Po wykonaniu operacji plik należy odłączyć od strumienia danych: `int fclose(FILE *ptr);`
- Funkcja **fclose()** wykonuje czyszczenie bufora.



# Odczyt i zapis plików

## **fgetc( )** i **fputc( )** znak-po-znaku

```
int fgetc(FILE *stream);  
int fputc(int c, FILE *stream);
```

## **fgets( )** i **fputs( )** wiersz-po-wierszu

```
char *fgets(char *s, int n, FILE *stream);  
int fputs(const char *s, FILE *stream);
```

## **fread( )** i **fwrite( )** blok-po-bloku

```
size_t fread(void *ptr, size_t size,  
             size_t n, FILE *str);  
size_t fwrite(const void *ptr, size_t  
             size, size_t n, FILE *str);
```





# Funkcje do obsługi plików (**stdio.h**)

- rewind( )** – przewija plik do początku
- ferror( )** – sprawdza sukces operacji na pliku
- remove( )** – usuwa plik
- rename( )** – zmienia nazwę pliku
- tmpfile( )** – otwiera plik tymczasowy
- tmpname( )** – generuje niepowtarzalną nazwę pliku
- access( )** – sprawdza istnienie pliku w systemie
- fseek( )** – wyszukiwanie danych w pliku
- feof( )** – wykrywa koniec pliku



# Inne funkcje do obsługi plików

- Podobne i rozszerzone funkcje obsługi pliku dostępne są też poprzez pliki nagłówkowe **io.h** , **dos.h**, lecz nie są one zdefiniowane w standardzie ANSI C.
- Użyteczne funkcje działające na katalogach z **dir.h**:

**chdir( )**

**mkdir( )**

**opendir( ), closedir( )**

– zmienia nazwę katalogu

– tworzy katalog

– otwiera i zamyka strumień  
związany z katalogiem

