

# Systemy Operacyjne - Operacje na plikach



Andrzej Stroiński

Institute of Computer Science  
Poznań University of Technology

1 październik, 2012

# Wprowadzenie do ANSI-C

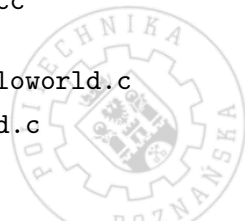
---

## Pomoc systemowa

- `man gcc`
- `man 2 write`
- `man 3 getc`

## Kompilacja programu

- `gcc <source_file>` np. `gcc helloworld.c`
- `gcc <source_file> -o <program_file>` np. `gcc helloworld.c -o helloworld`
- `gcc -Wall <source_file>` np. `gcc -Wall helloworld.c`
- `gcc -g <source_file>` np. `gcc -g helloworld.c`



# Wprowadzenie do ANSI-C

---

Brak opcji `-o` podczas kompilacji

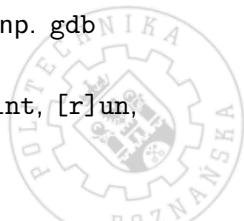
- program wykonywalny w pliku: `a.out`

Uruchomienie programu

- `<binary_file_path>` np. `/home/inf99999/helloworld` lub `./helloworld`

Debug w ANSI-C

- kompilacja programu z `-g`
- uruchomienie debugowania: `gdb <source_file>` np. `gdb helloworld`
- polecenie programu `gdb`: `[h]elp`, `[b]reak`, `[p]rint`, `[r]un`, `[c]ontinue`, `[q]uit`



# Wprowadzenie do ANSI-C - Hello World!

---

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

int main()
{
    printf("Hello_World\n");
};
```



## Wprowadzenie do ANSI-C - argumenty

---

```
#include <stdio.h>

int main(int argc, char* argv[])
{
    int i;
    for(i = 0; i < argc; i++)
    {
        printf("argument_%d: %s\n",
              i, argv[i]);
    }
    printf("Koniec ... \n");
    return 0;
}
```



## Wprowadzenie do ANSI-C - obsługa błędów

---

- Szczegółowy kod błędu można odczytać badając wartość globalnej zmiennej `errno` typu `int`.
- Obsługa błędów — funkcja `perror` (bada wartość zmiennej `errno` i wyświetla tekstowy opis błędu, który wystąpił).

```
int main(int argc , char* argv [])  
{  
    int fd=open("aaaa.txt" , O_RDONLY);  
    if (fd==-1)  
    {  
        printf("Kod:_%d\n" , errno);  
        perror("Otwieranie_pliku");  
    }  
    ...  
    return 0;  
}
```



# Operacje na plikach

---

## System operacyjny

- środowisko programów tworzących podstawową platformę programową dla działania innych aplikacji
- nadzoruje uruchomione programy oraz urządzenia sprzętowe komputera
- zarządza przydziałem pamięci operacyjnej, dysku, czasem procesora itd.

## Jądro systemu operacyjnego

- najważniejszy element systemu operacyjnego
- komunikacja ze sprzętem
- obsługa systemu plików (ext2, ext3, NTFS)
- sterowanie procesami



# Operacje na plikach

---

## System plików Linux

- **wszystko jest plikiem!** Katalog oraz urządzenie to też plik (specjalny)!
- plik to strumień bajtów, z którego można czytać bądź do którego można zapisywać. Zanim zostanie wykonana na pliku jakakolwiek operacja należy **plik otworzyć**, lub **utworzyć** jeśli jeszcze nie istnieje!
- w celu dostępu do pliku posługuje się jego nazwą (każdy plik może mieć wiele nazw - linki)

Plik może zostać otwarty w trybie:

- do zapisu - możliwa operacja tylko zapisu,
- do odczytu - możliwa operacja tylko do odczytu
- do zapisu i odczytu - możliwa operacja odczytu jak i zapisu





# Operacje na plikach

---

## Tablica i-węzłów

- przechowywana i utrzymywana przez jądro systemu operacyjnego
- zawiera informacje o i-węzłach (plikach):
  - typ pliku (dowiązanie symboliczne, zwykły plik, katalog, urządzenie znakowe, urządzenie blokowe, gniazdo, kolejka FIFO) i prawa dostępu
  - prawa dostępu
  - rozmiar
  - czas dostępu
  - licznik dowiązań (plik usuwany gdy równy 0)
  - flagi pliku (atrybuty "tylko dodawanie (append only)", "nie można zmieniać (immutable)", i inne)
  - ...



# Operacje na plikach - tłumaczenie nazwy na i-węzeł

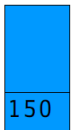
/home/pjackow/docs

Katalog główny

1	.
1	..
4	var
45	boot
56	home
9	tmp

i-węzeł home  
ma numer 56

i-węzeł 56



dane home  
znajdują się  
w węźle 150

Katalog home

5	.
1	..
11	ttomcz
23	kwojn
41	pjackow
34	fdomi

i-węzeł pjackow  
ma numer 41

i-węzeł 41

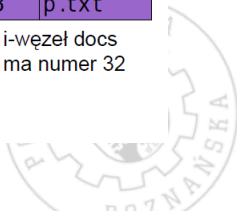


dane pjackow  
znajdują się  
w węźle 111

Katalog pjackow

7	.
5	..
69	temp
32	docs
64	test
73	p.txt

i-węzeł docs  
ma numer 32



## Operacje na plikach

---

Każdemu plikowi odpowiada jedna pozycja w tablicy **i-węzłów** w jądrze. Z każdym otwarciem pliku skojarzona jest pozycja w **tablicy otwartych plików** jądra. Ponieważ plik może zostać otworzony wiele razy, kilka pozycji w tablicy otwartych plików może wskazywać ten sam i-węzeł.

- plik jest identyfikowany poprzez **deskryptor** (indeks w tablicy deskryptorów)
- tablica deskryptorów zawiera wskaźnik do tablicy otwartych plików systemu (jądra)
- tablica deskryptorów ma ograniczony rozmiar
- przydzielany jest **zawsze wolny deskryptor o najmniejszej wartości** (wolną pozycję o najniższym numerze w tablicy deskryptorów)



# Operacje na plikach

---

W momencie utworzenia procesu otwarte są już i przypisane do tablicy deskryptorów następujące trzy pliki o deskryptorach związanych zazwyczaj z plikiem specjalnym terminala:

- 0 — standardowe wejście
- 1 — standardowe wyjście
- 2 — standardowe wyjście diagnostyczne

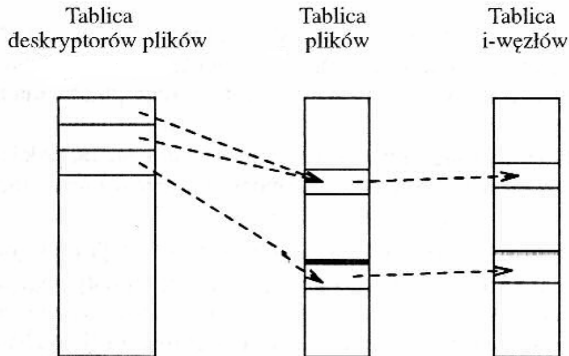
Funkcje tworzące i operujące na plikach opisane są w 2 części pomocy systemowej.



# Operacje na plikach

## Operacje na plikach

- Tablica deskryptorów pliku (proces)
- Tablica otwartych plików (globalna systemu operacyjnego)
- Tablica i-węzłów (jądro systemu)

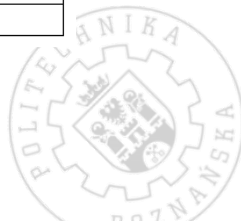


# Operacje na plikach

---

Przykładowa zawartości tablicy deskryptorów plików:

0	<code>stdin</code>	standardowe wejście
1	<code>stdout</code>	standardowe wyjście
2	<code>stderr</code>	standardowe wyjście diagnostyczne
3	<code>przykład.txt</code>	otwarty plik
4		
...	...	



Dziękuję za uwagę!

---

