

# Bartłomiej Długosz – jądro systemu na przykładzie LINUX

## LABORATORIUM

### WPROWADZENIE:

Poniższe zadanie składa się z dwóch części – w pierwszej należy napisać program, który będzie działał podobnie jak polecenie bashowe **find**. W drugiej części należy zapisać wynik wyszukiwania do pliku, a także zmienić jego prawa dostępu.

Cała trudność zadania polega na tym, że wszystko należy wykonać z poziomu programu, przy użyciu odpowiednich **wywołań systemowych (system calls)** oraz znalezieniu odpowiednich plików nagłówkowych.

Co do kwestii teoretycznej dotyczącej wywołań systemowych odsyłam do wideoprezentacji umieszczonej na dole – temat 6.

### WYMAGANIA:

**Docker** lub najnowsza wersja **gcc**

### PRZYGOTOWANIE:

a) Uruchom skrypt:

```
./run_dlugosz_bartlomiej.sh clone
```

b) Jeśli posiadasz gcc i chcesz pisać rozwiązanie na własnym komputerze, to pominiemy ten podpunkt.

Skrypt, który uruchamia środowisko (obraz dockera - zainstalowany **gcc** oraz **nano**):

```
./run_dlugosz_bartlomiej.sh run
```

Jeżeli występują jakiekolwiek errorry (permission denied):

```
sudo groupadd docker
```

```
sudo usermod -aG docker $USER
```

```
sudo chown "$USER":"$USER" /home/"$USER"/.docker -R
```

```
sudo chmod g+rwX "/home/$USER/.docker" -R
```

```
newgrp docker
```

Teraz ponownie uruchom skrypt:

```
./run_dlugosz_bartlomiej.sh run
```

# ZADANIE DO WYKONANIA

## CEL ZADANIA:

Wykorzystanie w praktyce wywołań systemowych do manipulowania plikami z poziomu programu - komunikacja pomiędzy jądrem systemu a programem.

## TREŚĆ ZADANIA:

### a) część pierwsza

Napisz program w języku C, który znajdzie i wypisze na standardowe wyjście wszystkie pliki oraz katalogi, które w swojej nazwie zawierają dany ciąg znaków (zadany jako argument wywołania programu).

Wyniki powinny być wypisane z widocznymi wcięciami, aby zaznaczyć ścieżkę, w której się znajdują, a katalogi oznaczone: [to\_jest\_katalog].

Kompilować do pliku wykonywalnego o nazwie 'my\_find'.

**WSKAZÓWKA:** pomocne będzie przestudiowanie manuala m.in. opendir, readdir, string.

### b) część druga

1) tym razem wynik z konsoli zapisz w buforze (zakładamy, że rozmiar nie przekroczy 1000 znaków).

2) następnie z **poziomu programu** stwórz plik o nazwie „result.txt”, który będzie miał następujące prawa dostępu: grupa oraz inni będą mieli prawo do odczytu, natomiast użytkownik ma posiadać prawa do odczytu oraz zapisu.

**Uwaga:** zabronione jest użycie funkcji:

```
int open(const char *pathname, int flags, mode_t mode);  
int open(const char *pathname, int flags);
```

Użyj odpowiedniego **system calla**, aby pozyskać informację na temat uprawnień do jego dostępu, które wypiszesz na konsolę.

3) teraz, przy użyciu **system calla** zmień prawa dostępu do pliku „result.txt” na 707. Ponownie wypisz jego prawa dostępu na konsolę.

4) następnie przywróć plik „result.txt” do praw dostępu z punktu 2), używając drugiego sposobu na zmianę a następnie wypisz tę informację na konsolę.

5) na koniec należy zamknąć plik, który otworzyliśmy oraz zwolnić wszelką zaalokowaną pamięć.

### Na koniec

Aby wyczyścić to, co zostało ściągnięte przez skrypt 'clone' należy użyć flagi 'clean'.

```
./run_dlugosz_bartlomiej.sh clean
```

**Link do wideoprezentacji:**

<https://www.youtube.com/watch?v=9qwW-VgKIz0>

**Materialy:**

<http://linasm.sourceforge.net/docs/syscalls/index.php>

<https://stackoverflow.com/questions>