# „Materiały wprowadzające”

dr inż. Arkadiusz Chrobot

22 lutego 2020

# Spis treści



# Wprowadzenie

W tych materiałach wstępnych zawarte są krótkie opisy narzędzi, których znajomość będzie niezbędna do sprawnej realizacji zadań w ramach laboratoriów z Systemów Operacyjnych 2. Każdemu zespołowi zostanie przypisana maszyna wirtualna, która trzeba będzie obsłużyć przy pomocy konsoli, dlatego należy przypomnieć sobie podstawowe polecenia powłoki systemowej. Te polecenie nie będą opisane w tych materiałach. W rozdziale 1 opisany jest sposób posługiwania się poleceniem `ssh` (ang. *Secure Shell*), w rozdziale 2 przedstawiono polecenie `scp` (ang. *Secure Copy*), a rozdział 3 poświęcony jest użytkowaniu serwisu Linux Cross Reference.

# 1. `ssh`

Polecenie `ssh` służy do zdalnej pracy z komputerem. Praca ta odbywa się za pomocą powłoki (konsoli). Pierwszą czynnością w tej pracy jest zalogowanie się do zdalnego systemu. Najpierw należy się zalogować do serwera maszyn wirtualnych jako użytkownik `so2lab`, wydając polecenie:

```
ssh so2lab@zefir.tu.kielce.pl
```

Zdalny system zapyta o hasło. Należy wprowadzić to, które poda osoba prowadząca laboratorium. Podczas pisania hasła na ekranie nie będą się pojawiały żadne znaki, do czasu naciśnięcia klawisza `Enter`. **Proszę uważnie wprowadzać hasło, bo zbyt duża liczba pomyłek może doprowadzić do czasowego zablokowania konta!!!** Po zalogowaniu się do serwera można zalogować się do maszyny wirtualnej. Maszyny wirtualne, które będą dostępne w ramach laboratoriów będą miały przypisany adres IP postaci `192.168.0.X`, gdzie `X` jest ostatnią liczbą adresu maszyny wirtualnej przypisanej do konkretnego zespołu. Aby się do niej zalogować na konto użytkownika `student` należy wydać następujące polecenie:

```
ssh student@192.168.0.X
```

Kiedy maszyna zapyta o hasło należy je wpisać, pamiętając, że w czasie wpisywania na ekranie nie będą pojawiały się żadne znaki[1].

Aby zalogować się do tej samej maszyny wirtualnej, ale na konto użytkownika `root` należy wydać następujące polecenie:

```
ssh root@192.168.0.X
```

Również w tym przypadku należy podać hasło, kiedy zdalny komputer o nie poprosi.

Istnieje również alternatywny sposób zalogowania się na konto użytkownika `root`. Należy najpierw zalogować się jako użytkownik `student`, a następnie wydać polecenie:

```
su -
```

Po wydaniu tego polecenia system zapyta o hasło dla użytkownika `root`. Należy je podać.

Trzeci sposób zalogowania się na konto użytkownika `root` jest bardzo podobny do drugiego. Pierwszym korkiem również jest zalogowanie do maszyny wirtualnej na konto `student`, ale po wykonaniu tej czynności należy wydać polecenie:

```
sudo su -
```

W tym przypadku system zapyta o hasło użytkownika `student`, a nie `root`.

[1] Hasła do poszczególnych kont zostaną podane na zajęciach.

## 2. `scp`

Polecenie `scp` należy do tego samego pakietu, co `ssh`. Umożliwia ono kopiowanie plików pomiędzy komputerem lokalnym, a zdalnym. Aby skopiować plik o nazwie `source.c` znajdujący się w bieżącym katalogu na serwerze maszyn wirtualnych, do katalogu domowego użytkownika `root` na maszynie wirtualnej o adresie IP `192.168.0.X` należy wydać następujące polecenie:

```
scp source.c root@192.168.0.X:~
```

Proszę zwrócić uwagę na znak dwukropka po adresie IP, po którym występuje nazwa katalogu. W tym wypadku jest to katalog domowy użytkownika, który oznaczany jest znakiem tyldy (~).

Analogicznie można skopiować pliki w odwrotnym kierunku lub między serwerem maszyn wirtualnych, a komputerem z którego jesteśmy zalogowani do tego serwera.

Jeśli chcielibyśmy skopiować cały katalog do zdalnego systemu, to mamy do dyspozycji dwa sposoby. Załóżmy, że ten katalog nazywa się `sources` i jest podkatalogiem bieżącego katalogu.

Pierwszy sposób polega na użyciu opcji `-r` polecenia `scp`, która powoduje rekurencyjne skopiowanie katalogu:

```
scp -r sources root@192.168.0.X:~
```

W drugim sposobie najpierw musimy ten katalog skompresować np. za pomocą polecenie `tar`. Możemy to zrobić następująco:

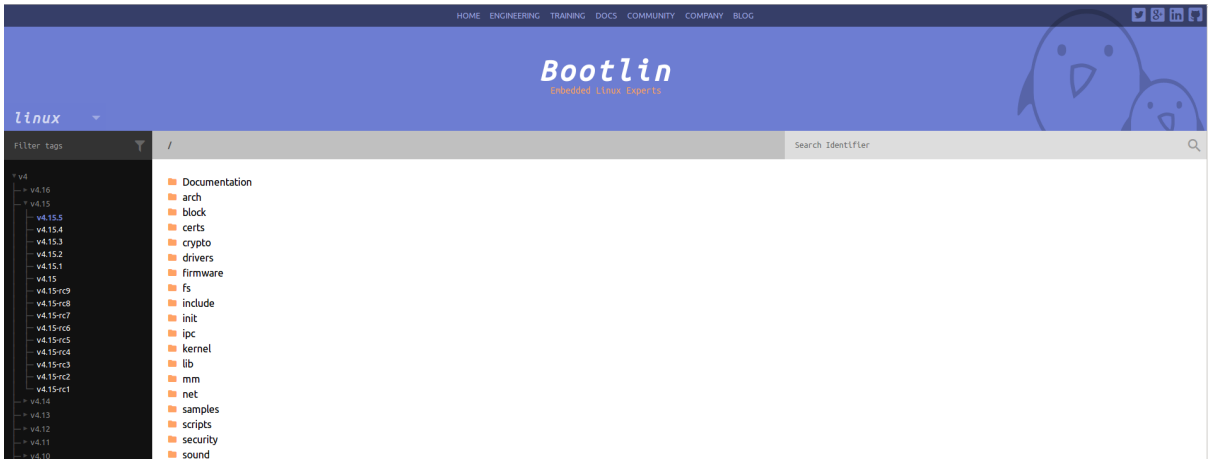
```
tar jcvf sources.tar.bz2 sources,
```

gdzie `sources.tar.bz2` jest nazwą pliku będącego skompresowanym archiwum katalogu `sources`. Po skopiowaniu go do zdalnego komputera możemy go rozpakować poleceniem:

```
tar xvf sources.tar.bz2
```

## 3. Linux Cross Reference

Istnieje dokumentacja do kodu źródłowego jądra Linuksa w postaci stron podręcznika `man`, ale niestety jest ona bardzo ograniczona. Najlepszym sposobem na zbadanie co dany fragment kodu robi i jak jest zbudowany jest przeglądanie źródeł jądra, które są bardzo obszerne. Na szczęście istnieje narzędzie, które potrafi wygenerować strony HTML pozwalające łatwo nawigować po tym kodzie. Tak opracowane strony dostępne są, między innymi, pod adresem `https://elixir.bootlin.com/linux/latest/source`. Poniżej opisany jest sposób korzystania z tego serwisu na przykładzie wyszukiwania informacji o funkcji `schedule()`[2]. Po wprowadzeniu wspomnianego adresu do przeglądarki i naciśnięciu klawisza `Enter`[3] powinniśmy zobaczy stronę, której fragment jest widoczny na rysunku 1.

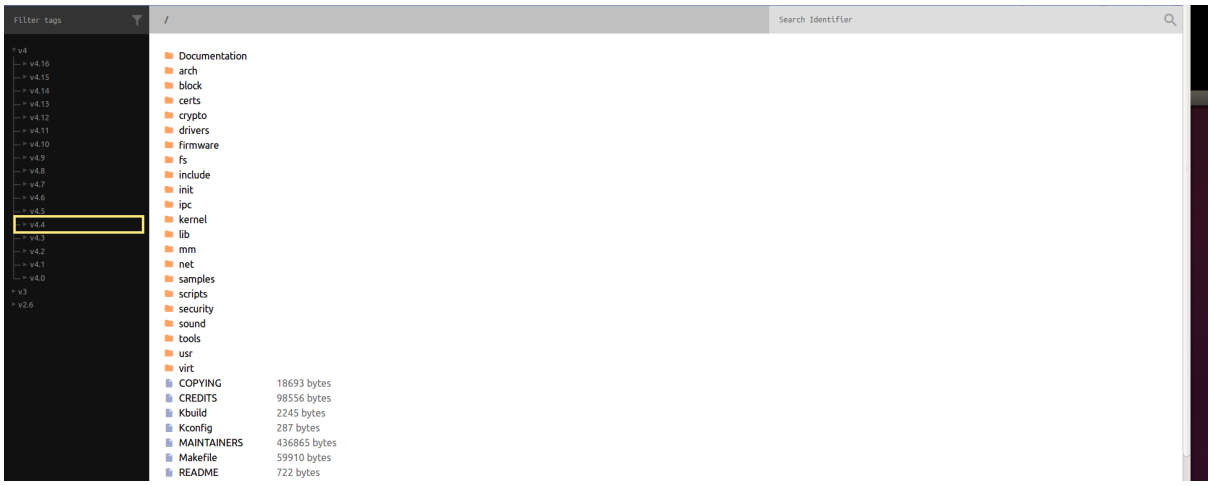



Rysunek 1: Strona główna serwisu Linux Cross Reference

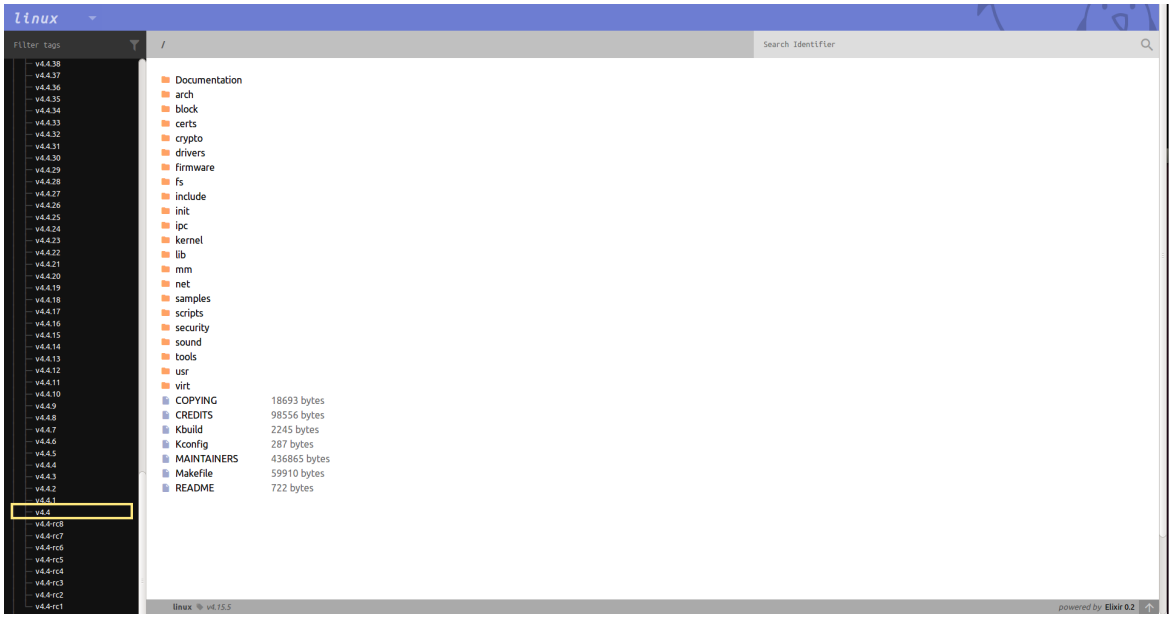
[2] Ta funkcja jest implementacją planisty procesora.

[3] Alternatywnie można kliknąć odnośnik na stronie przedmiotu.

Na tej stronie należy wybrać wersję `4.4` źródeł jądra, w sposób pokazany na rysunkach 2 i 3.

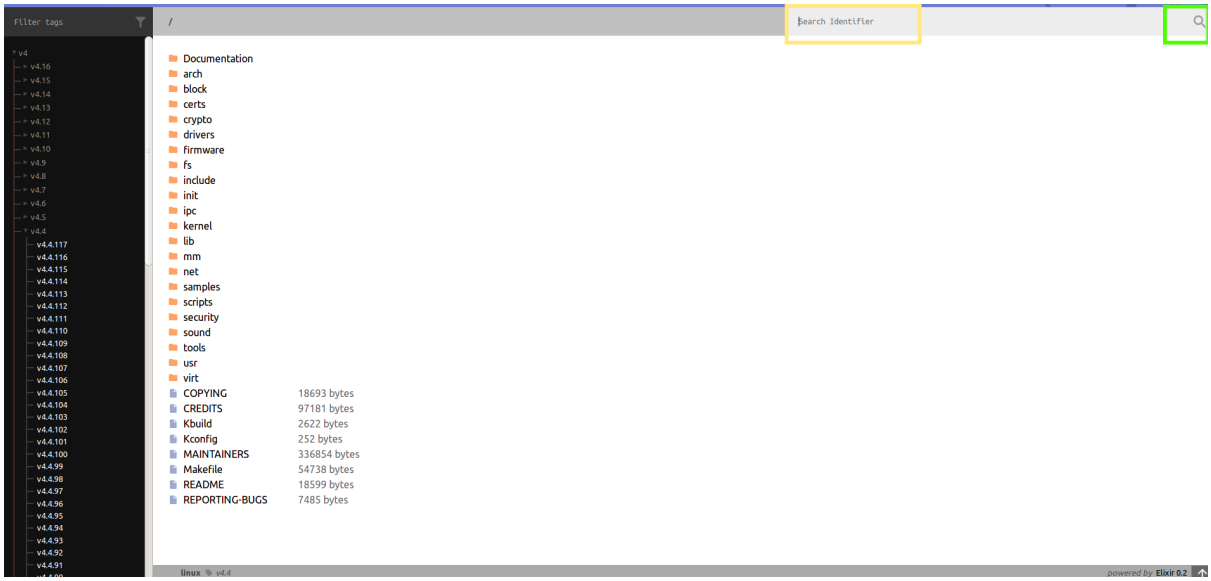



Rysunek 2: Menu wyboru wersji źródeł jądra




Rysunek 3: Szczegółowe menu wyboru wersji źródeł jądra

Po wybraniu wersji kodu źródłowego jądra należy w polu *Search Identifier*, wprowadzić nazwę *schedule*, a następnie kliknąć ikonę lupy lub nacisnąć klawisz `Enter` (rysunek 4).

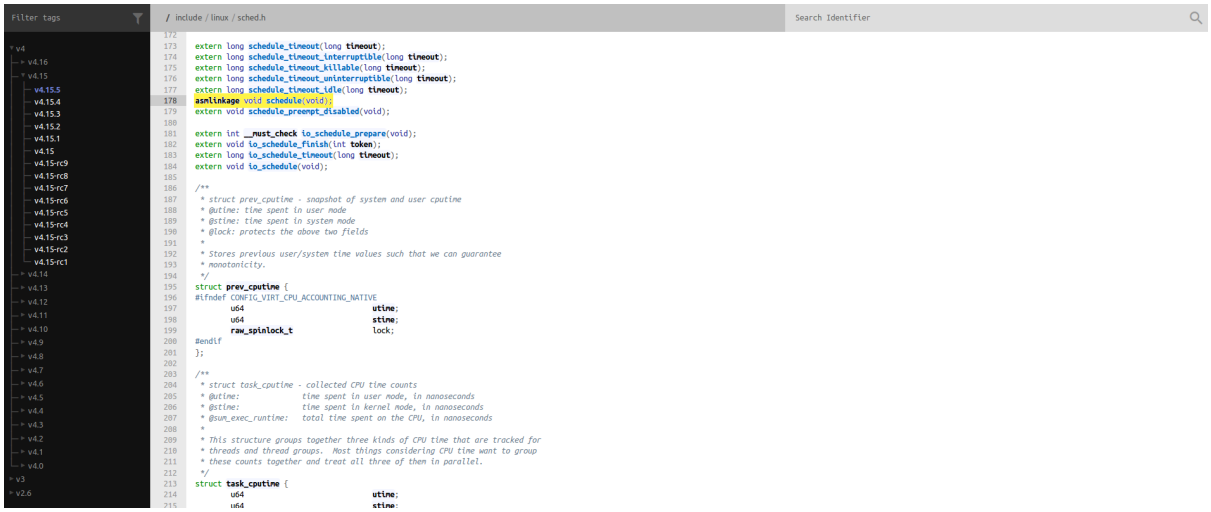



Rysunek 4: Wyszukiwanie nazwy

Strona z wynikami jest podzielona na dwie części. Pierwsza część, zaznaczona na rysunku 5 na żółto, zawiera odnośniki do wierszy w plikach, w których jest umieszczony prototyp lub definicja szukanej funkcji. Druga część, której fragment jest oznaczony kolorem zielonym, to lista odnośników do wierszy plików, w których ta funkcja jest używana.

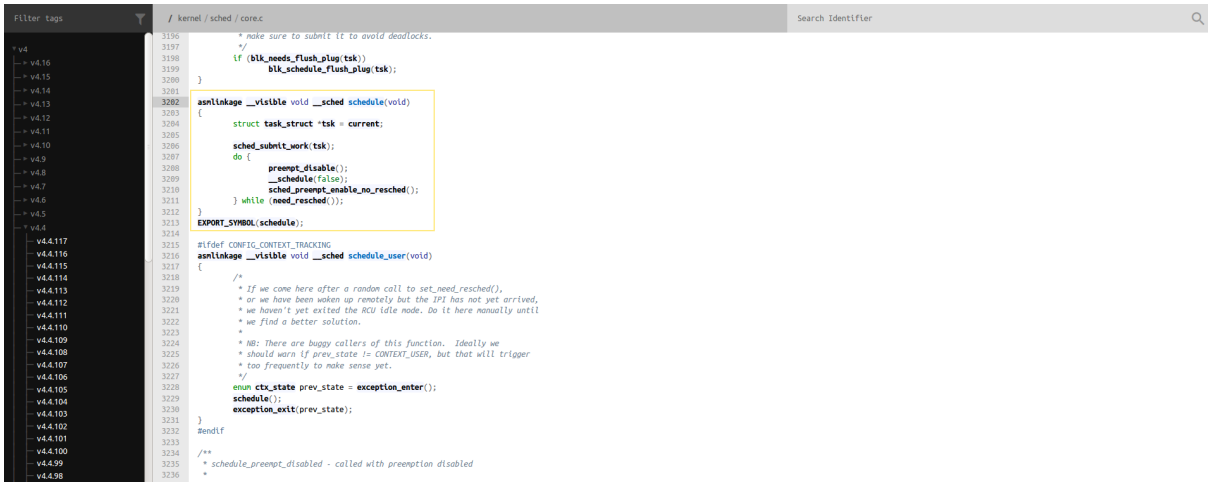

Rysunek 5: Wyniki wyszukiwania nazwy

Po kliknięciu w pierwszy odnośnik w sekcji *Defined in 3 files:* użytkownik zostanie przeniesiony do strony z deklaracją funkcji. Jej prototyp (nagłówek) został oznaczony na żółto na rysunku 6.




Rysunek 6: Deklaracja funkcji `schedule()`

Po kliknięciu w drugi odnośnik we wspomnianej sekcji użytkownik zostanie przeniesiony do strony z definicją tej funkcji, która została zaznaczona na żółto na rysunku 7.




Rysunek 7: Definicja funkcji `schedule()`