„Materiały wprowadzające”

dr inż. Arkadiusz Chrobot

24 lutego 2024

Spis treści

Wprowadzenie

W 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, którą trzeba będzie obsłużyć przy pomocy konsoli, dlatego należy przypomnieć sobie podstawowe polecenia powłoki systemowej. Te polecenia nie będą opisane w tych materiałach. W rozdziale 1 przedstawiony jest sposób posługiwania się poleceniem `ssh` (ang. *Secure Shell*), w rozdziale 2 opisano polecenie `scp` (ang. *Secure Copy*), a rozdział 3 poświęcony jest użytkowaniu serwisu Linux Cross Reference.

1. Polecenie `ssh`

Polecenie `ssh` służy do zdalnej pracy z komputerem. Praca ta odbywa się za pomocą powłoki systemowej (konsoli), a komunikacja między komputerem lokalnym i zdalnym jest szyfrowana. W przypadku zajęć laboratoryjnych tym zdalnym komputerem będzie maszyna wirtualna. Aby się z nią połączyć należy wydać na lokalnym komputerze, w terminalu lub konsoli, następujące polecenie:

```
ssh so2@81.26.20.32 -p X
```

W tym poleceniu `so2` to nazwa użytkownika, `81.26.20.32` to adres IP serwera maszyn wirtualnych, a `X` oznacza numer portu. Każdy zespół będzie miał przypisaną inną maszynę, z którą będzie się komunikował na określonym porcie. Numer tego portu będzie przekazany na pierwszych lub drugich zajęciach. Po nawiązaniu połączenia przez polecenie `ssh` zdalny system zapyta o hasło. Ono także zostanie przekazane w trakcie laboratorium. Podczas pisania hasła na ekranie nie będą się pojawiały żadne znaki, do czasu naciśnięcia klawisza `Enter`. Należy zaznaczyć, że z maszyną wirtualną można połączyć się tylko i wyłącznie z wewnętrznej sieci Katedry Systemów Informatycznych i jedynie wtedy, gdy ta maszyna jest uruchomiona.

Opisane polecenie pozwala zalogować się na konto użytkownika `so2`, które nie jest kontem uprzywilejowanym. Aby móc załadować lub usunąć moduły do jądra, co jest konieczne do realizacji zadań z instrukcji, należy skorzystać z konta użytkownika `root`. Można to zrobić następującym poleceniem:

```
su -
```

Po jego wydaniu system zapyta o hasło dla użytkownika `root`. Ono również zostanie przekazane przez osobę prowadzącą na zajęciach. Po podaniu prawidłowego hasła uzyskuje się dostęp do konta pozwalającego na administrację systemem, w tym wykonywanie wspomnianych operacji.

Inny sposób zalogowania się na konto użytkownika `root` polega na wydaniu polecenia:

```
sudo su -
```

W tym przypadku system zapyta o hasło użytkownika `so2`, a nie `root`. Jeśli będzie ono prawidłowe, to tak jak poprzednio otrzymuje się dostęp do powłoki systemowej tego ostatniego użytkownika.

Polecenie `sudo` ma też inne zastosowania. Ogólnie służy do uruchamiania poleceń uprzywilejowanych z poziomu użytkownika nieuprzywilejowanego. Nie mogą to być jednak dowolne polecenia. Muszą one być wcześniej określone przez administratora systemu dla tego użytkownika.

2. Polecenie `scp`

Polecenie `scp` należy do tego samego pakietu, co `ssh`. Umożliwia ono kopiowanie plików pomiędzy komputerem lokalnym i zdalnym przy użyciu szyfrowanej komunikacji. W przypadku laboratoriów, aby skopiować plik o nazwie `source.c` znajdujący się w bieżącym katalogu na komputerze lokalnym, do

katalogu domowego użytkownika `so2` na maszynie wirtualnej o porcie X należy wydać następujące polecenie:

```
scp -P X source.c so2@81.26.20.32:~
```

Proszę zwrócić uwagę na znak dwukropka po adresie IP, po którym występuje nazwa katalogu. W tym wypadku jest to katalog domowy użytkownika, który w Linuksie oznaczany jest znakiem tyldy (~). Jeśli miałby to być podkatalog katalogu domowego, np. o nazwie `lab1`, to należałby dokonać następującej zmiany w tym poleceniu:

```
scp -P X source.c so2@81.26.20.32:~/lab1
```

Podobnie jak w przypadku polecenia `ssh`, zdalny system zapyta o hasło użytkownika `so2`.

Analogicznie można skopiować pliki w odwrotnym kierunku, czyli np. aby skopiować plik `Makefile` znajdujący się w katalogu domowym użytkownika `so2`, na maszynie wirtualnej, do bieżącego katalogu na komputerze lokalnym, należy wydać następujące polecenie:

```
scp -P X so2@81.26.20.32:~/Makefile .
```

Kropka na końcu polecenia oznacza katalogu bieżący. Proszę zwrócić uwagę, że kolejność argumentów w `scp` jest taka sama, jak w zwykłym poleceniu `cp`, które kopiuje pliki lokalnie.

Aby skopiować cały katalog do zdalnego systemu, można użyć jednego z dwóch sposobów. W przykładach założono, że katalog do skopiowania nazywa się `sources`, jest podkatalogiem bieżącego katalogu i należy go umieścić w katalogu domowym użytkownika `so2` na maszynie wirtualnej.

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

```
scp -P X -r sources so2@81.26.20.32:~
```

W drugim sposobie najpierw należy ten katalog skompresować np. za pomocą polecenia `tar`. Można to zrobić następująco:

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

Plik o nazwie `sources.tar.bz2` będzie skompresowanym archiwum katalogu `sources`. Po skopiowaniu go do zdalnego komputera można go rozpakować poleceniem:

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

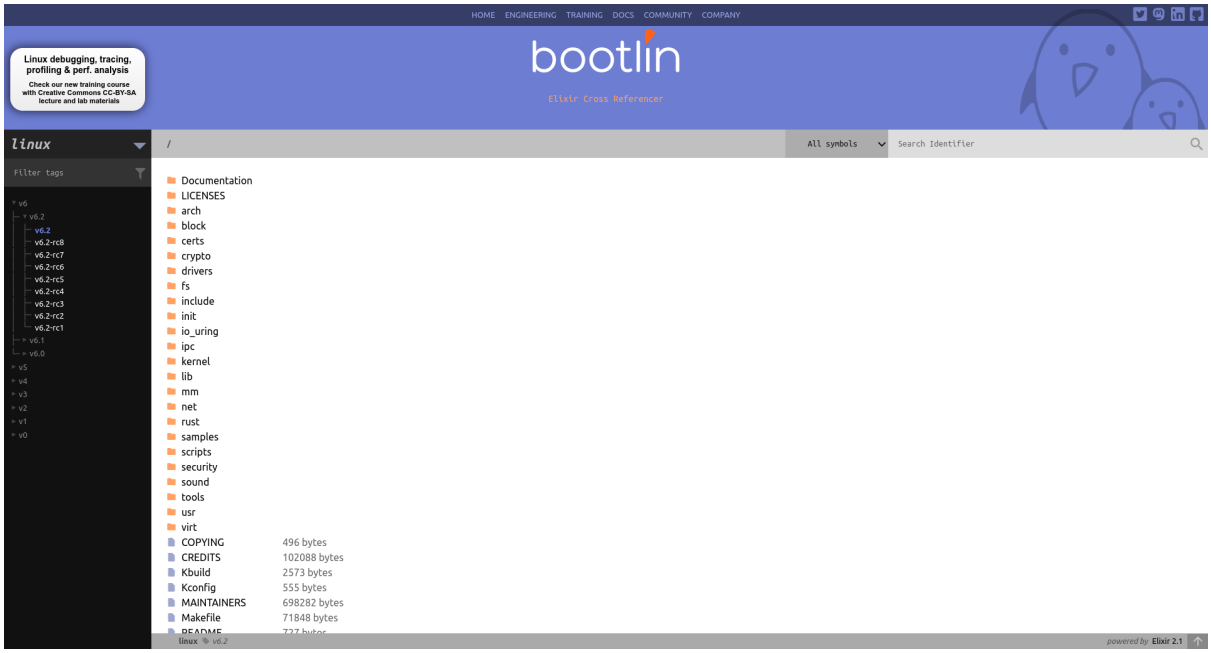
3. Linux Cross Reference

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 wygenerowane strony dostępne są, między innymi, pod adresem `https://elixir.bootlin.com/linux/latest/source`.

Sposób korzystania z nich zostanie opisany na przykładzie wyszukiwania informacji o definicji funkcji `schedule()`[1]. Po wprowadzeniu wymienionego adresu do przeglądarki i naciśnięciu klawisza `Enter`[2] pojawi się strona, której fragment jest widoczny na rysunku 1. Na tej stronie należy wybrać wersję `4.15` źródeł jądra, w sposób pokazany na rysunkach 2 i 3. 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). 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. Po kliknięciu w odnośnik w sekcji *Defined in 1 files as a prototype:* użytkownik zostanie przeniesiony do strony z deklaracją funkcji. Jej prototyp (nagłówek) został oznaczony na żółto na rysunku 6. Po kliknięciu odnośnika w sekcji *Defined in 1 file as a function:* (na rysunku 5 zaznaczona na zielono) użytkownik zostanie przeniesiony na stronę z definicją tej funkcji, która została zaznaczona na żółto na rysunku 7.

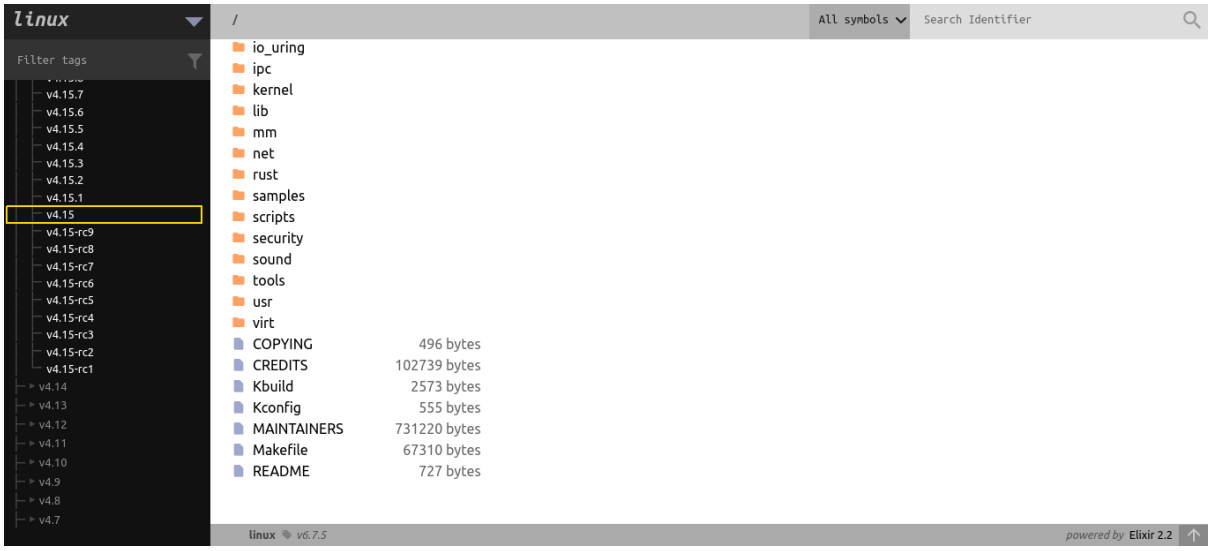
[1] Ta funkcja jest implementacją planisty procesora.

[2] Alternatywnie można kliknąć odnośnik na stronie przedmiotu.



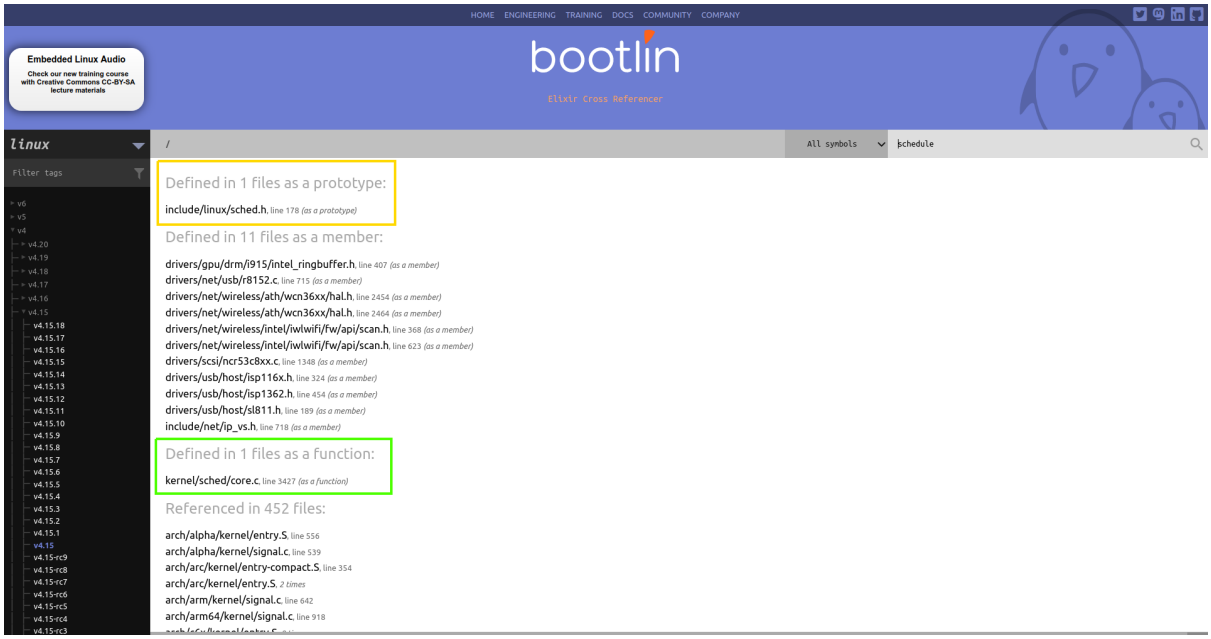
Rysunek 1: Strona główna serwisu Linux Cross Reference

Rysunek 2: Menu wyboru wersji źródeł jądra

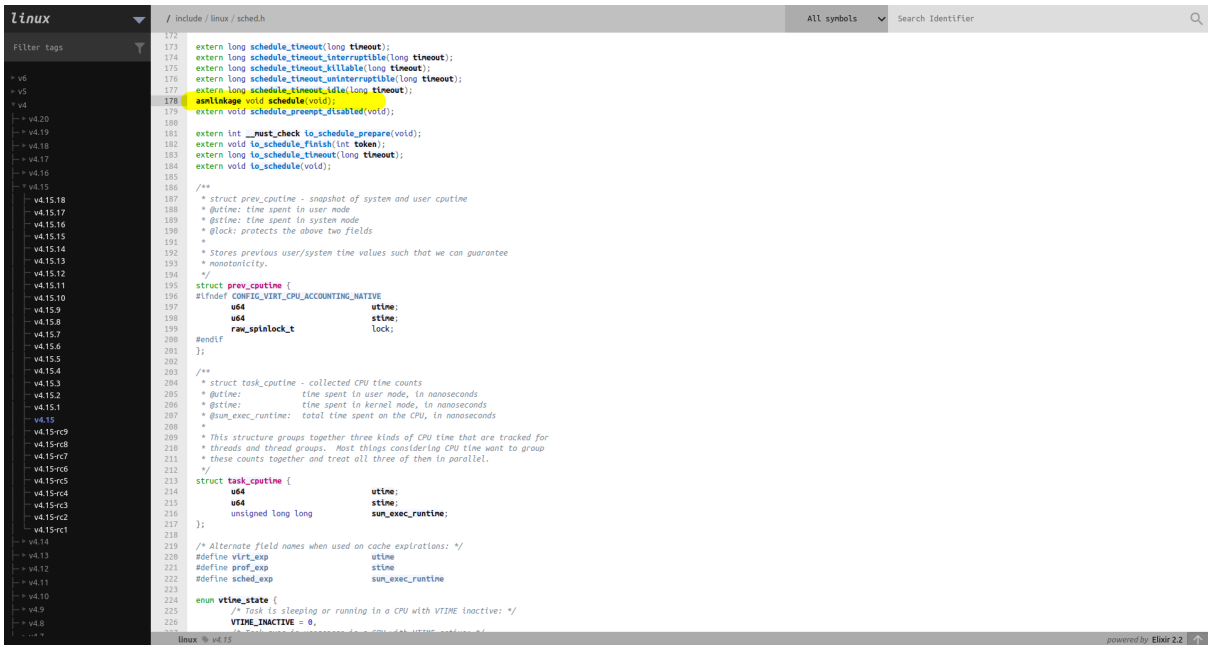


Rysunek 3: Szczegółowe menu wyboru wersji źródeł jądra

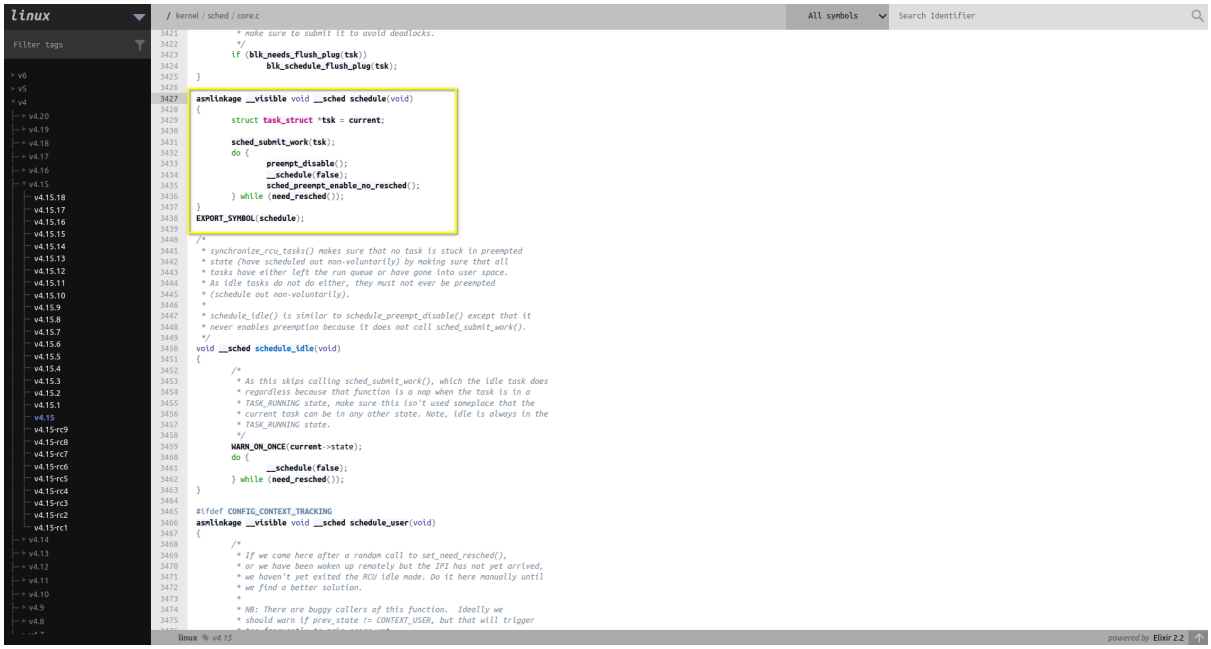
Rysunek 4: Wyszukiwanie nazwy



Rysunek 5: Wyniki wyszukiwania nazwy



Rysunek 6: Deklaracja funkcji `schedule()`



Rysunek 7: Definicja funkcji `schedule()`