Kompilacja jądra

Szymon Jędrych

1.06.2022

Linux

Przygotowanie plików jądra

Aby skompilować jądro, pierwszym krokiem jest pobranie aktualnej wersji. Na dzień 1 czerwca 2022 roku najnowszą wersją jest **5.18**. Jądro pobieramy ze strony https://kernel.org/. Wymienioną wersję pobieramy w archiwum o rozszerzeniu .tar ze strony https://cdn.kernel.org/pub/linux/k 5.18.tar.xz. Jeśli posiadamy wcześniej przygotowane jądro możemy wykorzystaćj je do kompilacji bez pobierania nowego.

Pliki jądra będziemy przechowywać w katalogu */usr/src*, dlatego przechodzimy tam komendą:

cd /usr/src

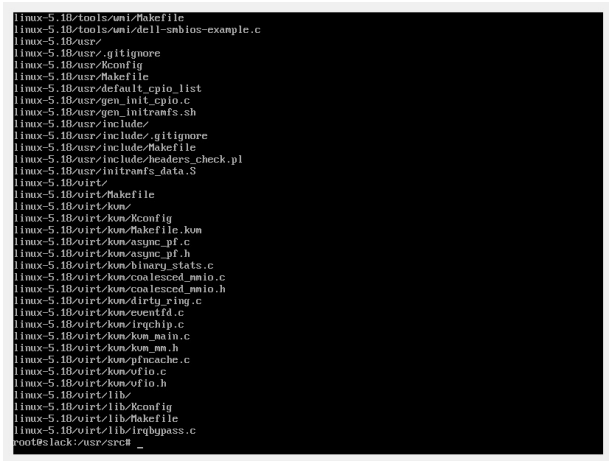
Po przejściu do podanego katalogu używamy polecenia *wget* do pobrania jądra linuxa:

wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.18.tar.xz

Rysunek 1: Pobieranie plików źródłowych

Następnie należy pliki wypakować poleceniem tar z argumentami -xvf:

tar -xvf linux-5.18.tar.xz

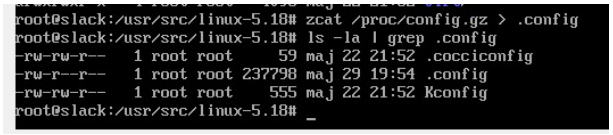


Rysunek 2: Wypakowywanie plików źródłowych

Po wypakowaniu plików jesteśmy gotowi do kompilacji jądra.

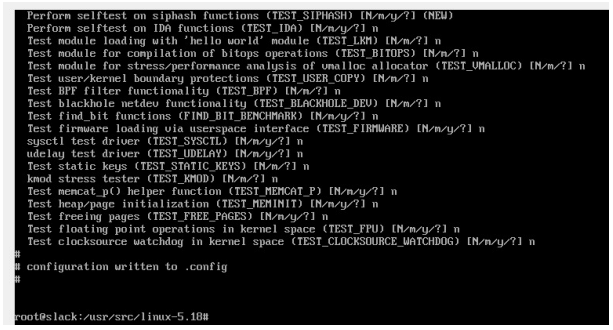
1 Metoda stara - tworzenie pliku konfiguracyjnego

Należy skopiować konfigurację aktualnego kernela do pliku .config



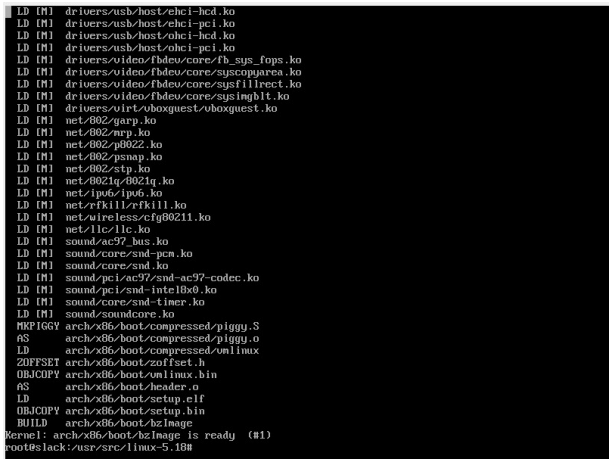
Rysunek 3: Skopiowanie aktualnego configu

Po utworzeniu configu używamy komendy *make localmodconfig* do stworzenia pliku konfiguracyjnego. Zostaniemy zapytani o to konfigurację poszczególnych modułów jądra, wszystko zostawiamy domyślnie. Wynik komendy:



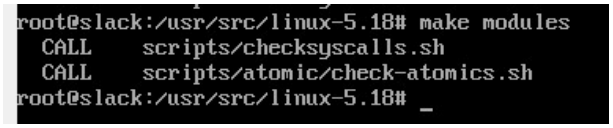
Rysunek 4: Zakończona konfiguracja jądra

Jesteśmy gotowi do kompilacji jądra. Użyta do tego została komenda *make*::



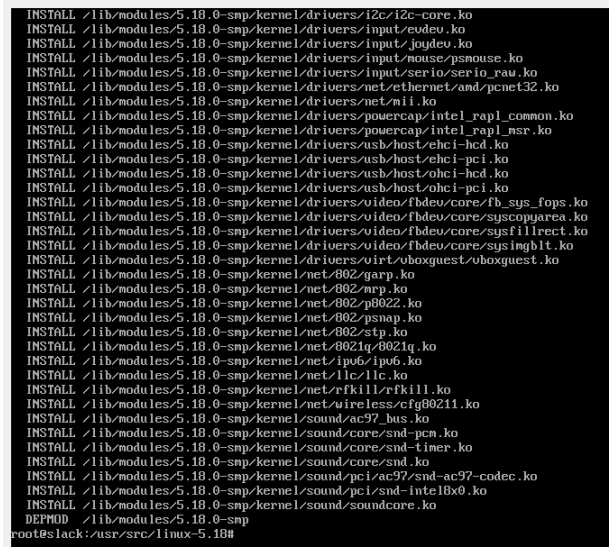
Rysunek 5: Zakończona kompilacja jądra

Po kompilacji jądra należy podobnie skompilować moduły komendą make modules:



Rysunek 6: Zakończona kompilacja modułów

Wykonujemy również komendę *make*
modules¡*nstallwceluwygenerowaniakomendygenerujcejramdisk* :

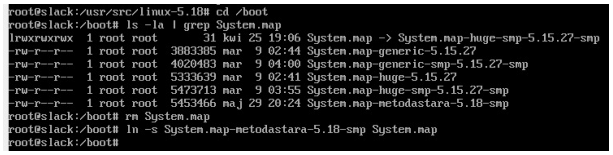


Rysunek 7: Komenda make install

Następnie należy skopiować pliki potrzebne do uruchomienia nowego jądra do katalogu /boot:

Rysunek 8: Kopiowanie plików do katalogu boot

Po kopiowaniu jesteśmy gotowi do zlinkowania pliku System.map:



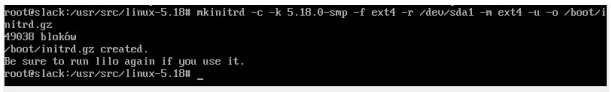
Rysunek 9: Podmienienie pliku System.map

Po linkowaniu możemy wygenerować komendę służącą do wygenerowania ramdisku:

Rysunek 10: Generowanie komendy do generacji ramdisk

Używamy teraz wygenerowanej komendy do utworzenia ramdisku:

mkinitrd -c -k 5.18.0-smp -f ext4 -r /dev/sda1 -m ext4 -u -o /boot/initrd.gz

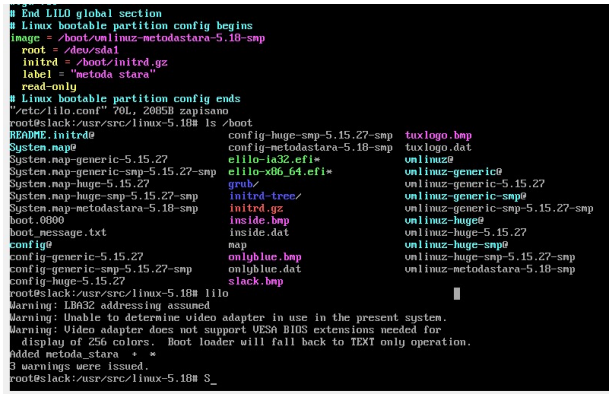


Rysunek 11: Generowanie ramdisk

Aby ukazać wygenerowany plik lilo.conf używamy cat /etc/lilo.conf:

Rysunek 12: Wygenerowany plik lilo.conf

Edytujemy plik w celu ustawienia nowego kernela i uruchamiamy komendę lilo:

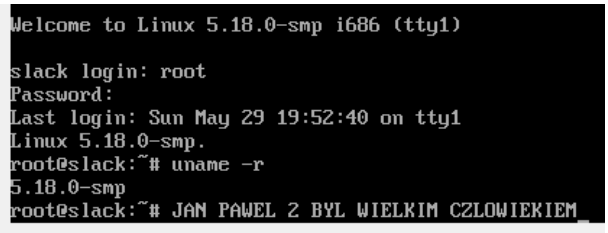


Rysunek 13: Edycja i uruchomienie lilo

Jesteśmy gotowi zrestartować maszynę. Wpis pojawił się przy uruchamianiu:

Rysunek 14: Uruchamianie nowego jądra

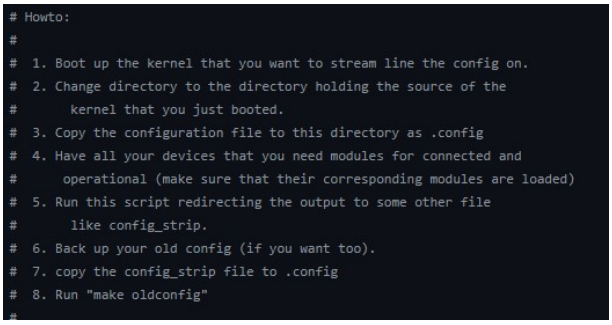
Po zalogowaniu widzimy, że nowe jądro zostało prawidłowo uruchomione:



Rysunek 15: Sprawdzenie działania jądra

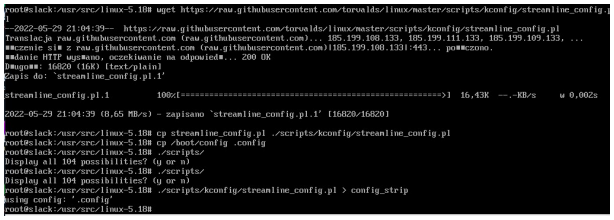
2 Metoda nowa - streamline$_config.pl$

Należy wygenerować plik konfiguracyjny zgodnie z instrukcją:



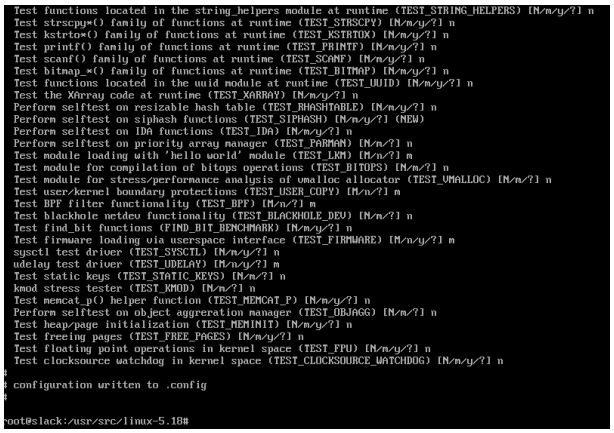
Rysunek 16: Instrukcja wygenerowania skryptu

Wykonujemy kilka komend niezbędnych do przygotowania środowiska:



Rysunek 17: Przygotowanie środowiska

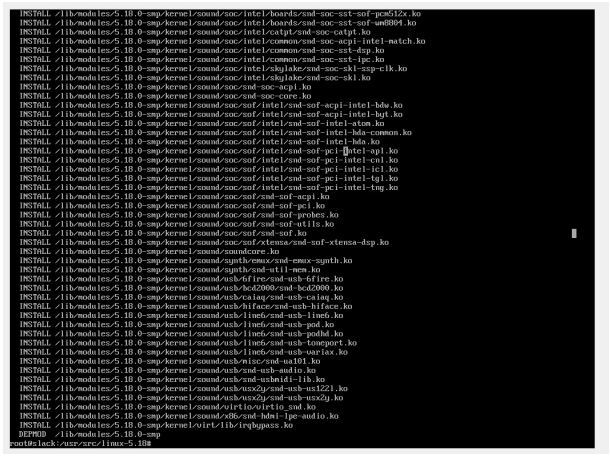
Po konfiguracji wykonujemy make oldconfig jak każe instrukcja:



Rysunek 18: Wykonanie komendy make oldconfig

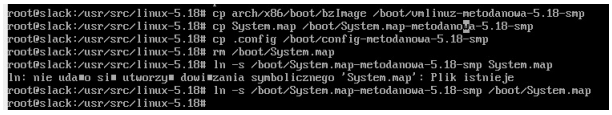
Następnie ponownie wykonujemy kompilację jądra i modułów oraz ich instalację komendą make make modules make modules_install

Rysunek 19: Kompilacja i instalacja jądra i modułów



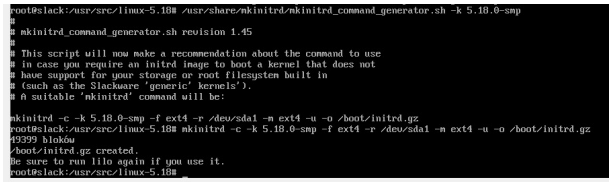
Rysunek 20: Wynik kompilacji i instalacji jądra/modułów

Wykonujemy następnie ten sam krop co w poprzedniej metodzie, to znaczy kopiujemy niezbędne pliki do katalogu /boot i tworzymy dowiązanie System.map:



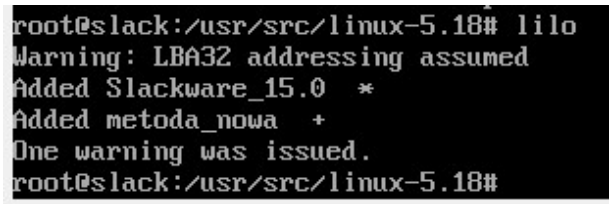
Rysunek 21: Przekopiowanie plików konfiguracyjnych i stworzenie dowiązań

Analogicznie tworzymy też ramdisk:



Rysunek 22: Stworzenie ramdisk

Dodajemy kolejny wpis oznaczający nową metodę do /etc/lilo.conf

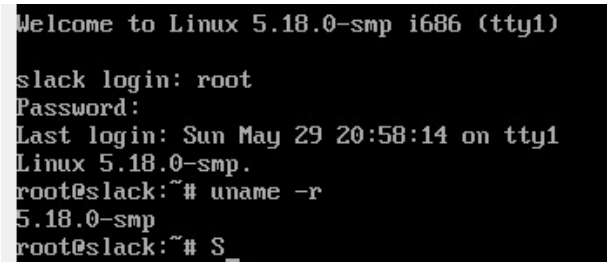


Rysunek 23: Nowe wejście w lilo.conf

Jesteśmy gotowi zrestartować maszynę i sprawdzić działanie nowego jądra. Nowy kernel pojawił się w menu wyboru:

Rysunek 24: Nowe jądro w oknie wyboru

System uruchomił się i działa poprawnie. Sprawdzamy wersję kernela aby upewnić się, że uruchomiliśmy prawidłową wersją komendą uname -r:



Rysunek 25: Test nowego jądra

3 Podsumowanie

Kompilacja jądra obydwoma sposobami była podobna. Zdecydowanie bardziej jednak podobała mi się kompilacja drugim, nowszym sposobem. Była odrobinę prostsza, trwała jednak kilka razy dłużej mimo tych samych ustawień maszyny wirtualnej.

Jedynym napotkanym problemem był problem z podaniem wersji kernela przy generowaniu komendy do utworzenia ramdisku. Okazało się, że przy podawaniu wersji należało podać ją

z 0 na końcu, tj. 5.18.0 zamiast samego 5.18. Był to problem stosunkowo łatwy do rozwiązania, wystarczyło spojrzeć na wynik kompilacji jądra.