## Lista 5 (Lab) Termin wysłania na SVN do 24.11.2019

1. *(5pt)* Napisz program w języku C, który uruchomi powłokę (Bash) z prawami roota. Po kompilacji programu można ustawić (z poziomu roota) dowolne atrybuty (np. patrz SUID). Następnie już z poziomu dowolnego użytkownika uruchamiając program uruchamia się konsola administratora, podobnie jak **sudo /bin/bash** (bez wprowadzania hasła).
2. *(5pt)* Napisz w języku C programy testujące, które odpowiedzą na następujące pytania:
   - Czy można napisać program do obsługi wszystkich sygnałów (patrz **kill -l**)? Napisz program prezentujący odpowiedź.
   - Czy jest możliwe wysłać sygnał **SIGKILL**, lub inny do procesu **init** (PID 1) czyli np. **kill -9 1** (nawet będąc **root**em)?
   - Czy sygnały są kolejkowane? Np. napisz program testowy wysyłający wiele razy do danego procesu sygnał (np. SIGUSR1) i zobacz czy wszystkie dotarły.
3. *(5pt)* Zaimplementuj w języku C prostą wersję powłoki o nazwie **lsh**. Jak prawdziwa powłoka, **lsh** odczytuje linię ze standardowego wejścia i przeszukuje ścieżki ze zmiennej **PATH** (inaczej mówiąc zamiast **execve** wykonuje **execvp**) i wykonuje podany program. Proszę pamiętać o ustawieniu argumentów wykonywanej komendy. Jeśli linia kończy się znakiem (&), wtedy **lsh** powinien nie czekać aż komenda zostanie skończona i od razu wrócić. W innym przypadku **lsh** powinien zaczekać, aż program wykona się. **lsh** powinien skończyć swoje działanie naciskając klawisze **Control+D** lub pisząc **exit**. Zmieniamy katalogi przez wpisanie komendy **cd**. Komendy **cd** oraz **exit** to komendy wbudowane. **Uwaga:** Procesy uruchomione w tle, które się zakończyły mogą stać się procesami 'zombi', rozwiąż ten problem w lsh.
4. *(10pt)* Zaimplementuj w programie **lsh** z poprzedniego zadania potoki **|** (ang. pipe) oraz przekierowanie standardowego wejścia (**<**), wyjścia (**>**) oraz wyjścia błędu (**2>**). Wskazówka: Zobacz program lssort.c. Ponadto Ctrl-C przerywa wykonywanie programu w powłoce (nie samej powłoki oraz zadań wykonywanych w tle).
5. *(15pt)** Zaimplementuj w programie **lsh** zarządzanie zadaniami (job-control). Patrz książka Michael Kerrisk, "Linux Programming Interface".