

WPROWADZENIE DO UNIX'A / LINUX'A (PROCESY I SIĘĆ)

1. Procesy

- Każdy proces ma swój unikalny w zakresie całego systemu numer identyfikacyjny zwany `PID` (ang. process identification number). Oprócz numeru `PID` każdy proces (z jednym wyjątkiem) ma numer `PID` swojego rodzica - tzw. `PPID` (ang. parent `PID`), czyli numer `PID` procesu, przez który został uruchomiony.
- Jedynym procesem, który nie ma swojego rodzica, jest proces o nazwie `init`, który jest jako pierwszy uruchamiany po starcie systemu operacyjnego. Proces `init` jest przodkiem wszystkich innych procesów.
- Aby mieć kilka procesów do manipulowania, wykonaj polecenie: `sleep 15m`
- Zaloguj się w nowym terminalu.
- Użyj polecenia `write twój_login`. (Jeśli trzeba, włącz możliwość rozmowy `mesg y`). Aby wstrzymać działanie programu nie przerywając go całkiem wciśnij kombinację klawiszy `CTRL-Z`.
- Z poziomu terminala można uruchamiać zadania działające w tle, tzn. takie, których działanie nie powoduje zawieszenia działania terminala. Do uruchamiania zadań w tle służy symbol "&" umieszczany na końcu polecenia. Wpisz: `pine &`
- Polecenie `ps` generuje listę wszystkich aktywnych procesów w systemie, ich stan, rozmiar, nazwę oraz właściciela. Istnieje bardzo dużo opcji tego polecenia (patrz `man ps`). My skorzystamy z `ps -ef`. Prawdopodobnie lista ta będzie dłuższa niż liczba linii terminala. Aby wyświetlić tę listę z podziałem na strony, wpisz polecenie `ps -ef | more`

```
regulski@tempus:~$ ps -ef
UID          PID  PPID  C  STIME TTY          TIME CMD
regulski    7718   7715  0  13:38 ?          00:00:00 sshd: regulski@pts/1
regulski    7719   7718  0  13:38 pts/1      00:00:00 -bash
regulski    8126   7719  0  13:52 pts/1      00:00:00 write regulski
regulski    8249   8246  0  13:57 ?          00:00:00 sshd: regulski@pts/2
regulski    8250   8249  0  13:57 pts/2      00:00:00 -bash
regulski    8263   8250  0  13:57 pts/2      00:00:00 pine
regulski    8301   8250  0  13:58 pts/2      00:00:00 write regulski
regulski    8302   8250  0  13:58 pts/2      00:00:00 sleep 15m
regulski    8303   7719  0  13:58 pts/1      00:00:00 ps -ef
```

- Znaczenie poszczególnych kolumn jest następujące:
 - `UID` - nazwa użytkownika (właściciela procesu)
 - `PID` - numer identyfikacyjny procesu
 - `PPID` - numer `PID` rodzica danego procesu
 - `C` - miara obciążenia procesora
 - `STIME` - czas rozpoczęcia działania procesu
 - `TTY` - nazwa terminala związanego z procesem
 - `TIME` - czas wykorzystania procesora
 - `CMD` - komenda, która wywołała proces
- Aby wyświetlić wszystkie procesy tylko jednego użytkownika, wpisz polecenie `ps -fu nazwaużytkownika`
- Aby zabić jakiś proces należy posłużyć się programem `kill`. Generalnym przeznaczeniem tego programu jest wysyłanie sygnałów do uruchomionych procesów. Składnia polecenia jest następująca (gdzie `-n` określa numer sygnału):


```
kill -n PID
```

Najważniejsze sygnały to:

 - `-1` - (hang up) zawies.
 - `-9` - (kill) zabij.
 - `-15` - wstrzymaj proces.

Jeśli wydamy polecenie `kill PID` to do procesu zostanie wysłany sygnał `-15`.
- Aby zamknąć wszystkie procesy danego użytkownika zastosuj:


```
killall -u user
```
- W systemach unixowych zwykły użytkownik (nie będący administratorem) może manipulować jedynie procesami, które do niego należą. Aby się o tym przekonać w ramach samodzielnych eksperymentów, spróbuj zabić jakiś dowolny proces, którego nie jesteś właścicielem.

2. Co jeszcze oferuje nam **Unix**

- Niniejsze ćwiczenie miało na celu zaprezentowanie jedynie podstawowych zagadnień związanych z procesami w systemie **Unix**. System ten oferuje użytkownikom znacznie większe możliwości zarządzania procesami, np:
 - pozostawianie działających procesów nawet po wylogowaniu się z systemu (polecenie `nohup`)
 - uruchamianie zadań w zadanej chwili (polecenia `crontab` oraz `at`)
 - określanie priorytetów procesów (polecenie `nice`).

3. WinSCP

- Podstawową wadą protokołu transmisji plików **FTP** jest brak wbudowanych mechanizmów bezpieczeństwa, a zwłaszcza szyfrowania transmisji. Zarówno nazwa użytkownika, hasło jak i dane, wędrując siecią narażone są na możliwość ich podsłuchania. Tej wady pozbawione są protokoły **SCP** (ang. secure copy). Jest to godna zalecenia metoda transmisji plików poprzez sieć Internet.
- Aplikacja **WinSCP** umieszczona jest na twoim komputerze w systemie **Windows**. Pozwala na połączenie z serwerem, na którym zainstalowany jest **Unix** i zapewnia graficzny interfejs dla obsługi **scp**. Pliki na swoim koncie za serwerze **tempus.metal.agh.edu.pl** możesz umieszczać i przysyłać na inne komputery za pośrednictwem tej aplikacji.
- Program **WinSCP** umożliwia wykonywanie wszystkich podstawowych operacji na plikach (zarówno w obrębie lokalnego oraz zdalnego komputera): kopiowanie lub przenoszenie plików, tworzenie katalogów, zmiany nazw, usuwanie plików lub katalogów oraz zmiana uprawnień do plików na zdalnym komputerze.
- Większość operacji można wykonać przy pomocy klawiszy funkcyjnych opisanych na dolnej belce okna programu. Wszelkie operacje wykonywane są wyłącznie w aktywnym panelu, a w przypadku operacji kopiowania i przenoszenia - z panelu aktywnego do nieaktywnego. Operacje kopiowania i przenoszenia można również wykonywać myszką korzystając z techniki "ciągnij i upuść".
- Powyżej paneli znajdują się dwa małe poziome okienka umożliwiające wybór bieżącego katalogu na lokalnym i zdalnym komputerze. Po prawej stronie każdego z nich znajduje się zestaw ikon służących do zmiany bieżącego katalogu na katalog nadrzędny, katalog główny (**root**), katalog domowy oraz do odświeżenia zawartości panelu. Nazwy plików i katalogów ukrytych wyświetlane są w kolorze szarym (przypomnienie: w systemie **Unix** jako ukryte traktuje się te pliki, których nazwa zaczyna się od kropki). Linki symboliczne na zdalnym komputerze oznaczane są podobnie jak skróty w systemie Windows - poprzez umieszczenie w lewym dolnym rogu ikony pliku, miniaturowego kwadratu z symbolem strzałki.
- Stwórz w systemie Windows pusty plik tekstowy o nazwie **<twoje nazwisko>.txt**
- Uruchom teraz program **WinSCP**, uruchom połączenie ze swoim kontem na **tempusie**. Stwórz na swoim koncie katalog **public_html**, a następnie spróbuj przesłać do niego plik tekstowy z twojego komputera.
- Zmień prawa dostępu dla katalogu i pliku (prawy przycisk myszki, **Properties**) na prawo wykonywania i odczytu dla wszystkich.
- Edytuj plik i umieść w nim napis „**Hello World!**”
- W przeglądarce podaj adres pliku (będzie to wyglądało mniej więcej tak):
<http://tempus.metal.agh.edu.pl/~regulski/regulski.txt>

4. DNS

- Host to komputer podłączony do Internetu lub innej sieci używającej protokołu `TCP/IP` i posiadający unikalny adres `IP`.
- Hosty identyfikują się za pomocą adresów `IP`. Jednak adresy `IP` są trudne do zapamiętania przez ludzi i głównie z tego powodu został wprowadzony system nazw domen `DNS` (ang: Domain Name System), który kojarzy nazwy domen z odpowiadającymi im adresami `IP`.
- Do konwersji nazw `DNS` na adresy `IP` i vice-versa służy polecenie `host`.
- Polecenie `host` służy także do innych celów - np. do znajdowania hosta, który obsługuje daną domenę jako serwer poczty.
- Przykład:
`host tempus.metal.agh.edu.pl`
- Wpisując to polecenie dostaniemy numer `IP` naszego serwera.
- Polecenie: `/usr/bin/net status sessions` wyświetla aktywne połączenia protokołu `TCP`

5. Polecenie `ping`

- Polecenie `ping` służy do sprawdzania, czy dany host w sieci jest osiągalny z naszego komputera i pozwala także na zmierzenie czasów odpowiedzi.
- Sprawdź, czy serwer `galaxy.uci.agh.edu.pl` jest dostępny z sieci:
`ping galaxy.uci.agh.edu.pl`
(po kilku `ping`-ach) `CTRL+C`