Laboratorium 3: Procesy.

Wstęp: Konfiguracja stanowiska.

Ćwiczenia można przeprowadzić na przykładzie każdego systemu Unix.
Część ćwiczeń wymaga uprawnień użytkownika root.
Ćwiczenia są przygotowane na przykładzie systemu Knoppix 3.6 uruchomianego z płyty CD.
W przypadku prowadzenia zajęć na innej wersji systemu Unix prowadzący poinformuje o istotnych różnicach w ćwiczeniach.

Wymagania minimalne:
- Komputer klasy PC ,
- Procesor Intel Celeron 366,
- Pamięć RAM 20 MB,
- Napęd CD-ROM.
- BIOS umożliwiający uruchomienie komputera z płyty CD.

UWAGA! Wszelkie zmiany wprowadzone do systemu plików i konfiguracji systemu operacyjnego znikają po restarcie komputera, ponieważ system jest uruchamiany z płyty CD i system plików nie jest zapisywany na dysku.

Przygotowanie komputera:
- Uruchom komputer.
- Włóż do napędu CD-ROM płytę zatytułowaną „Knoppix 3.6”.
- Zaczekaj na komunikat określający sposób uruchomienia programu konfigurującego BIOS komputera.
- Uruchom program konfigurujący BIOS komputera.
- Odszukaj w konfiguracji komputera parametr umożliwiający uruchomienie systemu operacyjnego (boot from CD-ROM) z płyty CD.
- Ustaw opcję uruchamiania systemu z płyty CD.
- Zapisz konfigurację.

Rysunek 1 Uruchamianie systemu Knoppix.

- Zaczekaj komputer się uruchomi i pojawi się ekran powitalny systemu (Rysunek 1)
- Wpisz knoppix 2
- Naciśnij Enter.
- Zaczekaj na pojawienie się znaku zachęty.

Czynności wstępne.

Poniższe czynności należy wykonać tylko dla systemu Knoppix uruchomionego z płyty CD.

Aby uruchomić demona at:

- wpisz polecenie `atd` i naciśnij Enter.

Aby założyć dodatkowych użytkowników:

- utwórz katalog user1 w katalogu /ramdisk
- utwórz katalog user2 w katalogu /ramdisk
- nadaj do utworzonych katalogów uprawnienia tylko dla właścicieli
- wpisz polecenie `useradd -d /ramdisk/user1 -s /bin/bash -u 1001 user1` i naciśnij Enter.
- wpisz polecenie `useradd -d /ramdisk/user2 -s /bin/bash -u 1002 user2` i naciśnij Enter.
- zmień odpowiednio właścicieli dla utworzonych katalogów: userx dla katalogu userx.

Procesy w systemie UNIX

Proces jest to polecenie (program) w trakcie wykonywania. System UNIX jest systemem wielozadaniowym i wielodostępnym, co oznacza możliwość jednoczesnego wykonywania wielu procesów przez wielu użytkowników. Uruchomione polecenie (program) jest wstawiane do kolejki procesów przeznaczonych do wykonania i wykonywane w

przydzielonych mu kwantach czasu. Podziału czasu procesora w systemie UNIX powoduje przełączanie się pomiędzy wykonywanymi zadaniami. Kolejne procesy są pobierane z kolejki, wykonywane przez określony czas i odkładane z powrotem do kolejki.

Użytkownik systemu może wykonywać kilka procesów jednocześnie: jeden *proces pierwszoplanowy* i wiele innych *procesów w tle*.

Proces pierwszoplanowy to proces aktualnie wykonywany na terminalu (jest to zatem jedyny proces akceptujący wejście z klawiatury). Podczas działania procesu pierwszoplanowego nie jest możliwe użycie terminala do innych zadań. Uruchomienie innego zadania możliwe jest dopiero po zakończeniu procesu pierwszoplanowego.

Proces w tle to proces znajdujący się w kolejce procesów, ale nie kontrolowany z terminala (nie jest powiązany z terminalem). Dlatego wejście z klawiatury nie może być przekazane do procesu działającego w tle. Jako procesy działające w tle wykonuje się najczęściej zadania czasochłonne lub nie wymagające wprowadzania danych z klawiatury (stosuje się potoki lub strumienie).

Ćwiczenie 1. Informacje o wykonywanych procesach (polecenie ps)

Z każdym procesem wykonywanym w systemie związany jest szereg informacji. Najważniejsze z nich to identyfikator procesu PID oraz identyfikator procesu nadrzędnego PPID. Procesy połączone są w grupy. Grupę tworzy np. powłoka wraz ze wszystkimi uruchomionymi przez nią procesami. Proces, który uruchomił procesy danej grupy to tzw. lider grupy procesów (w tym przypadku np. proces powłoki).
Informacje o stanie wykonywanych procesów uzyskujemy poleceniem `ps`. Należy mieć na uwadze, że polecenie `ps` podaje przybliżone informacje o procesach, ponieważ zmieniają się one podczas działania polecenia.

Zadanie 1-1. Aby wyświetlić listę procesów użytkownika uruchomionych na bieżącym terminalu:

- wpisz polecenie `ps` i naciśnij Enter.
  Przykładowy wynik działania polecenia `ps`:

```
PID          TTY        TIME       CMD
24988        tty1       0:00       bash
25941        tty1       0:00       ps
```

gdzie:
`PID` – unikalny identyfikator procesu (użyteczny np. do przerwania procesu)
`TTY` – identyfikator terminala, z którego uruchomiono proces
`TIME` – całkowity czas wykonywania procesu
`CMD` – nazwa procesu (polecenia)

Jakie procesy są uruchomione?

Zadanie 1-2. Aby wyświetlić listę procesów użytkownika `user1` uruchomionych na wszystkich terminalach:

- wpisz polecenie `su - user1` i naciśnij Enter.
- naciśnij Alt-F2
- wpisz polecenie `ps -u user1` i naciśnij Enter.

Jakie procesy są uruchomione?

Zadanie 1-3. Aby wyświetlić dodatkowe informacje o procesach

- wpisz polecenie `ps -f` i naciśnij Enter.

Przykładowy wynik działania polecenia `ps -f`:

```
UID    PID   PPID   C   STIME     TTY  TIME   CMD
ciwe  27667  27665  0  15:51:31  tty1  0:01  -tcsh
ciwe  29361  27667  0  17:32:21  tty1  0:00  ps -f
```

gdzie:

UID – nazwa użytkownika, który uruchomił proces
PID – identyfikator procesu
PPID – identyfikator procesu nadrzędnego (rodzica, ang. *parent*)
C – wykorzystanie procesora na zarządzanie procesem
STIME – czas uruchomienia procesu
TTY – identyfikator terminala
TIME – czas wykonywania procesu
CMD – nazwa procesu (polecenia)

Zadanie 1-4. Aby wyświetlić listę wszystkich procesów w systemie za wyjątkiem liderów grup oraz procesów nie związanych z żadnym terminalem:

- wpisz polecenie `ps -a` i naciśnij Enter.

Zadanie 1-5. Aby wyświetlić listę wszystkich procesów działających w systemie:

- wpisz polecenie `ps -e` i naciśnij Enter.

UWAGA: (znak ? w kolumnie `TTY` oznacza procesy systemowe nie związane z żadnym terminalem)

Zadanie 1-6. Aby wyświetlić PID aktywnej powłoki:

- wpisz polecenie `echo $$` i naciśnij Enter.

Ćwiczenie 2. Uruchamianie procesów

(znak `&`, polecenia `at`, `batch`, `nohup`)

Polecenia wydawane z klawiatury są wykonywane według kolejności ich wprowadzania, jako procesy pierwszoplanowe, czyli w sposób asynchroniczny. Wykonywanie takiego polecenia musi zostać zakończone, zanim rozpocznie się wykonywanie następnego polecenia. Umieszczanie poleceń w tle pozwala wykonywać je w sposób synchroniczny. Po umieszczeniu polecenia (lub kilku poleceń) w tle, można w dalszym ciągu wykonywać polecenia w pierwszym planie, nawet jeśli wykonywanie poleceń w tle nie zostało jeszcze zakończone. Polecenie umieszcza się w tle podając na jego końcu znak `&`.

UWAGA: Po zakończeniu sesji przy terminalu wszystkie procesy w tle są przerywane. Nie dotyczy to jednak procesów zleconych do wykonania poleceniami `at`, `batch` i `nohup`. Wykonywanie procesów uruchomionych w ten sposób nie wymaga obecności użytkownika w systemie.

Ogólny zapis: `polecenie &`

Przykład 1: polecenie wykonywane jako proces pierwszoplanowy:

```
find / -name math.h -print | grep -v "find:" > wynik
```

Działanie polecenia (jako procesu pierwszoplanowego):

a) wyszukanie w katalogu głównym / i jego wszystkich podkatalogach (czyli w całym systemie plików) pliku o nazwie `math.h` (parametr `-name math.h`);
b) wypisanie ścieżek do znalezionych plików (parametr `-print`);
c) odfiltrowanie komunikatów o braku dostępu za pomocą polecenia `grep` (czyli uwzględnienie wyłącznie komunikatów nie zawierających ciągu znaków „`find:`") (parametr `-v „find:"`);
d) zapisanie wyników do pliku o nazwie `wynik` (przekierowanie strumienia wyjściowego z ekranu monitora do pliku `> wynik`).

Przykład 2: polecenie wykonywane w tle:

```
find / -name math.h -print | grep -v "find:" > wynik &
```

Zadanie 2-1. Wpisz polecenie z przykładów i zanotuj jakie są różnice?

Zadanie 2-2: Zlecanie zadań o określonej porze (polecenie at):

UWAGA: Jeżeli ćwiczenia są wykonywane w innym systemie niż Knoppix, to uruchamianie poleceń `at` i `batch` wymaga przydzielenia uprawnień przez administratora. Nadanie uprawnień polega na dopisaniu nazwy użytkownika uprawnionego do wykonywania tych poleceń do pliku `/usr/lib/cron/at.allow` lub usunięciu nazwy użytkownika z pliku `/usr/lib/cron/at.deny`.

UWAGA: (tylko Knoppix) Ćwiczenie należy wykonać będąc zalogowynym jako użytkownik user1. Przełącz się na pierwszą sesję wciskając Alt-F1..

Aby zlecić wykonanie zadania:

- sprawdź aktualny czas systemowy.
- wpisz polecenie `at teraz+2` i naciśnij Enter (teraz oznacza czas systemowy z poprzedniej czynności).
- wpisz polecenie `mkdir at1` i naciśnie Enter.
- wciśnij ^D (Ctrl-D)

Zaczekaj dwie minuty i sprawdź czy został utworzony katalog at1.

Jaka jest data utworzenia katalogu?

W jakim katalogu został utworzony katalog at1?

Zleć do wykonania następujące zadania (zanotuj numery przydzielane zadaniom):

- za 2 minutu ls –l >at2
- za 10 minut mkdir at3
- za 20 minut ls
- za 40 minut find / -name „at" –print > at5

Aby wyświetlić listę zadań oczekujących na wykonanie:
- wpisz polecenie `at -l` i wciśnij Enter.
- wpisz polecenie `atq` i wciśnij Enter.

Aby usunąć zlecone zadanie:
- wpisz polecenie `at -l` i wciśnij Enter.
- określ numer zadania do usunięcia.
- wpisz polecenie `at -d <numer zadania>` i wciśnij Enter.

W podobny sposób jak polecenie `at` działa polecenie `batch` z tą różnicą, że proces nie jest wykonywany w zadanym czasie, tylko w momencie małego obciążenia systemu.

Zadanie 2-3: Powtórz zadania używając polecenia batch zamiast at.

Jakie były różnice?

Zadanie 2-4: Utworzenie skryptu pomocniczego.

Utwórz plik s1 zawierający poniższy skrypt i nadaj mu prawo wykonywania:

```
#!/bin/bash
while true
do
        echo $0 - `date`
        sleep 15
done
```

UWAGA: Zwróć uwagę na znaki ` w poleceniu echo. Są to znaki akcentu (na klawiaturze znaku 1) a nie apostrofy (na klawiaturze obok L).

Zadanie 2-5: Uruchamianie zadania w tle i wylogowanie się.

Aby uruchomić zadanie w tle:
- wpisz polecenie `./s1 &` i wciśnij Enter.
- zaczekaj i upewnij się, że co 15 sekund na ekranie pojawia się aktualny czas i data.
- wpisz polecenie `ps -ef` i wciśnij Enter.
- zanotuj PPID powłoki wykonującej skrypt s1.
- wpisz polecenie `exit` i wciśnij Enter.
- zaczekaj i sprawdź czy aktualny czas i data nadal pojawiają się na ekranie.
- wpisz polecenie `exit` i wciśnij Enter.
- zaczekaj i sprawdź czy aktualny czas i data nadal pojawiają się na ekranie.
- wpisz polecenie `ps -ef` i wciśnij Enter.
- zanotuj PPID powłoki wykonującej skrypt s1.

UWAGA: W systemach innych niż Knoppix nastąpi wylogowanie użytkownika. W celu dokończenia zadania należy ponownie się zalogować.

Czy zanotowane PPID są identyczne? Dlaczego?

Czy aktualna data i czas nadal pojawiają się na ekranie?

Zadanie 2-6: Uruchamianie zadań, które nie będą zakończone po wylogowaniu się użytkownika.

Powtórz poprzednie zadanie używając polecenie `nohup s1` zamiast ./s1 &

UWAGA: W przypadku braku przekierowania wyjścia do pliku efekty działania polecenia zapisywane są w katalogu użytkownika w pliku `nohup.out`.

UWAGA: Polecenie `nohup` nie współpracuje z potokami i listą poleceń – należy w takich przypadkach użyć skryptu.

Czy na ekranie pojawiała się data i czas?

Czy wielkość pliku nohup.out zmienia się?

Cwiczenie 3. Przerywanie procesów (polecenie `kill`)

Użytkownik może przerywać działanie wyłącznie własnych procesów. Procesy innych użytkowników może przerywać tylko root.
Proces pierwszoplanowy można zakończyć za pomocą kombinacji klawiszy CTRL–C.
Procesy w tle przerywamy za pomocą polecenia `kill`. Poleceniem `kill` można wysłać do procesu odpowiedni sygnał:

```
kill [nr_sygnału] <PID>
```

Domyślnie wysyłany jest sygnał `15` (`SIGTERM`). Jeżeli proces nie reaguje na sygnał SIGTERM, to stosuje się sygnał 9 (`SIGKILL`).

Przykłady:

`kill 0`	przerwanie wszystkich procesów wykonywanych w tle (za pomocą domyślnego sygnału `SIGTERM` o numerze 15)
`kill -9 0`	j.w., ale włącznie z przerwaniem procesu powłoki, co spowoduje wyjście z systemu
`kill -9 24988`	„zabicie” procesu o identyfikatorze `24988` za pomocą sygnału `SIGKILL` nr `9`

Zadanie 3-1: Zabij proces uruchomiony w zadaniu 2-6.

ĆWICZENIE 4: Reagowanie na sygnały w skryptach.

Polecenie wewnętrzne powłoki `trap` pozwala określić zachowanie się skryptu w reakcji na sygnały wysyłane poleceniem `kill`.

Składnia: `trap [polecenie] [n]`

W przypadku otrzymania sygnału `n` wykonywane jest `polecenie`. Puste polecenie `''` powoduje ignorowanie sygnału.

Utwórz plik s2 zawierający poniższy skrypt i nadaj mu prawo wykonywania:

```
#!/bin/bash
trap '' 15
while true
do
        echo $0 - `date` - Sprobuj mnie wylaczyc
        sleep 15
done
```

Zadanie 4-1: Aby zatrzymać proces pierwszoplanowy.

- uruchom skrypt s2 wpisując polecenie ./s2 i naciśnij Enter.
- Zaczekaj aż na ekranie pojawi się kilka komunikatów skryptu i naciśnij ^C (Ctrl-C).

Czy skrypt zakończył swoje działanie?

Zadanie 4-2: Aby zatrzymać proces nie reagujący na SIGTERM.

- uruchom skrypt s2 w tle wpisując polecenie ./s2 & i naciśnij Enter.
- Zaczekaj aż na ekranie pojawi się kilka komunikatów skryptu.
- Wyślij do skryptu sygnał SIGTERM.

Czy skrypt zakończył swoje działanie?

- Wyślij do skryptu sygnał SIGKILL.

Czy skrypt zakończył swoje działanie?

Ćwiczenie 5: Sterowanie wykonywaniem procesów. (polecenia `jobs`, `fg`, `bg`, `stop`)

Wszystkie procesy (zadania) mogą znajdować się w jednym z trzech stanów:

1. Zadanie znajdujące się w stanie *foreground* (proces pierwszoplanowy) ma możliwości odczytu i zapisu na związanym z tym zadaniem terminalu.
2. Zadanie znajdujące się w stanie *background* (proces w tle) nie ma możliwości odczytu i ma warunkową możliwość zapisu na związanym z tym zadaniem terminalu.
3. Zadanie znajdujące się w stanie *suspended* (lub *stopped*, proces wstrzymany) jest zadaniem wstrzymanym zazwyczaj w reakcji na odpowiedni sygnał.

Każde zadanie uruchomione w tle (proces w tle) otrzymuje oprócz identyfikatora `PID` identyfikator `job_id` w postaci dodatniej liczby całkowitej.

Utwórz kopie skryptu s1 o nazwach s3, s4, s5.

Zadanie 5-1: Aby wyświetlenie listę zadań pracujących w tle.

- uruchom skrypty s1,s3,4 w tle.

- wpisz polecenie `jobs` i naciśnij Enter.

Jakie procesy pracują w tle i jaki jest ich stan?

Zadanie 5-2: Aby przesunąć proces pierwszoplanowy w tło.
- uruchom skrypty s5 jako proces pierwszoplanowy.
- zaczekaj aż pojawią się komunikaty procesu s5.
- naciśnij ^Z (Ctrl-Z).

Jakie procesy pracują w tle i jaki jest ich stan?

Jakie komunikaty pojawiają się na ekranie?

Zadanie 5-3: Aby wznowić wykonywanie zadania w tle.
- wpisz polecenie `bg PID` i naciśnij Enter. (w miejsce PID wpisz pid skryptu s5)

Jakie procesy pracują w tle i jaki jest ich stan?

Zadanie 5-4: Aby uaktywnić jako pierwszoplanowy proces pracujący w tle.
- wpisz polecenie `fg PID` i naciśnij Enter. (w miejsce PID wpisz pid skryptu s1)

Jakie procesy pracują w tle i jaki jest ich stan?
