

Laboratorium

Procesy i zadania w Linuksie

I. Zadania

Zadania (ang. jobs) to programy uruchamiane w sesji przez powłokę. Terminy zadanie i proces są w literaturze SO używane prawie zamiennie. Operacje na zadaniach w systemie Linux:

Ctrl+Z	usypianie zadań (ang. suspend)
jobs	spis zadań
fg	przełączanie zadań
bg	praca w tle
kill	usuwanie zadań
& czyli ampersand	równoległe uruchamianie zadań

II. Procesy

Proces jest wykonywanym programem. Wykonanie procesu musi przebiegać w sposób sekwencyjny. Oznacza to, że w dowolnej chwili na zamówienie danego procesu może być wykonywany co najwyżej jeden rozkaz kodu programu

W wielozadaniowym systemie operacyjnym może działać jednocześnie wiele procesów.

Podstawowe zadania związane z zarządzaniem tymi procesami obejmują:

- tworzenie i usuwanie procesów,
- wstrzymywanie i wznawianie procesów,
- planowanie kolejności wykonania procesów,
- dostarczanie mechanizmów synchronizacji i komunikacji procesów.

Każdy proces charakteryzuje się pewnymi atrybutami :

przestrzeń adresowa, licznik programu, licznik stanu, licznik rejestru, deskryptory pliku (unikalny identyfikator pliku wykorzystywany przez system operacyjny), dane procesu, zależności rodzinne, liczniki statyczne.

Wynikiem obecności w systemie procesów jest to, że jądro systemu może nim sterować

Stany procesu:

pracujący w trybie użytkownika (proces znajduje się na procesorze i wykonuje kod),

pracujący w trybie jądra (jądro wykonuje wywołanie systemowe, wykonane przez proces)

uśpiony (proces czeka na jakieś zdarzenie, np. na odczyt danych z dysku lub otrzymanie danych z sieci), **gotowy do wykonania** (może być uruchomiony w każdej chwili, jednak nie ma jeszcze przydzielonego procesora), **zombie** (proces zakończył działanie i czeka na odebranie kodu powrotu przez proces macierzysty),

Podstawowym poleceniem do zarządzania procesami przez użytkownika jest : ps.

ps [-] [lujsvmaxscewhrnu] [txx] [0 [+/-] k1 [[+/-] k2 ...]] [pids]

np.: ps -A (Sun), ps -ax (Linux), a także ps -u user, ps aux|grep user

Program ps akceptuje kilka rodzajów opcji:

1 Opcje w stylu UNIX, które mogą być grupowane i muszą być poprzedzone myślnikiem.

2 Opcje BSD, które mogą być grupowane i nie mogą być użyte z myślnikiem.

3 Długie opcje GNU, które są poprzedzone dwoma myślnikami.

Różne typy opcji można ze sobą dowolnie mieszać, jednakże mogą pojawić się konflikty.

Aby zobaczyć wszystkie procesy w systemie, używając standardowej składni:

ps -e
ps -ef
ps -eF
ps -ely
ps axu

Aby wypisać drzewo procesów:

ps -ejH
ps axjf

Aby wyświetlić informację o wątkach:

ps -eLf
ps axms

Aby wyświetlić informacje związane z bezpieczeństwem:

ps -eo euser,ruser,suser,fuser,f,comm,label
ps axZ
ps -eM

Aby zobaczyć wszystkie procesy poza tymi działającymi jako root (rzeczywiste i efektywne ID):

ps -U root -u root -N

Wyświetlenie deskryptorów pól AIX:

ps -o "%u : %U : %p : %a"

Wyświetlenie tylko nazwy procesu o identyfikatorze 42:

ps -p 42 -o comm=

Definicje niektórych opcji:

-a - wyświetla wszystkie procesy posiadające terminal kontrolny, nie tylko bieżące procesy użytkowników.
-r - wyświetla tylko pracujące procesy.
-x - wyświetla procesy nie posiadające terminala kontrolnego.
-u - wyświetla właścicieli procesów.
-f - wyświetla powiązania między procesami - format drzewiasty.
-l - generuje listę w długim formacie.
-w - wyświetla parametry linii poleceń procesu (do połowy linii).
-ww - wyświetla wszystkie parametry linii poleceń procesu niezależnie od długości.
-j - format prac: pgid sid.
-s - format sygnału.
-m - wyświetla informacje o pamięci.
-S - dodaj czas CPU potomka i błędy stron.
-h - bez nagłówka.
-n - wyjście numeryczne dla USER i WCHAN.
-tXX - tylko procesy z kontrolującym tty XX

Definicje nagłówków:

USER - kto jest właścicielem danego procesu.

PID - numer identyfikacyjny procesu.

%CPU - procentowe wykorzystanie procesora.

%MEM - procentowe wykorzystanie pamięci przez proces.

VSZ - wielkość pamięci wirtualnej przydzielonej procesowi.

RSS - wielkość rzeczywistej pamięci wykorzystywanej przez proces.

TTY - terminal kontrolny procesu. ? w tej kolumnie oznacza, że proces nie jest połączony z żadnym terminalem kontrolnym.

STAT - stan procesu:

S - proces jest uśpiony.

R - proces jest aktualnie wykonywany.

D - proces jest uspiony, oczekuje na operacje (zazwyczaj I/O).

T - proces jest debuggowany lub został zatrzymany.

Z - zombie.

START - czas lub data rozpoczęcia procesu.

TIME - przedział czasu wykorzystany przez CPU.

COMMAND - nazwa procesu oraz jego parametry

Polecenie **top** jest interaktywną wersją polecenia **ps u -e**. Zamiast wyświetlać statyczną listę procesów w systemie, **top** odświeża listę co 2,3 sekundy (przedział czasu można ustawić samemu). Główną niedogodnością tego programu jest znaczne obciążenie CPU.

Obok nadawanego przez system priorytetu, każdy proces posiada wartość **nice**, którą użytkownicy mogą modyfikować. Użytkownicy mogą sterować wartością **nice** wykonywanych przez nich zadań, jednak tylko administrator może nadać tej wartości wartość ujemną. Aby uruchomić dowolny program z zadaną wartością "nice" na **nice** posłużyć się komendą **nice** w następującej składni:

nice [-PRIORYTET] KOMENDA

PRIORYTET - wartość "nice" nadawana komendzie. Należy poprzedzić ją myślnikiem.

wartość domyślna - priorytet +10.

KOMENDA - polecenie do wykonania.

Można również zmienić priorytet działającego już procesu. Do tego celu służy komenda **renice**.

renice PRIORYTET PID(Y)

PRIORYTET - oznacza nową wartość **nice**. W przypadku zwykłych użytkowników może to być wyłącznie wartość dodatnia i większa od obecnej,

PID(Y) - identyfikator(y) procesów.

Aby zabić jakiś proces należy posłużyć się program **kill**. Generalnym przeznaczeniem tego programu jest wysłanie sygnałów do uruchomionych procesów. Składnia polecenia jest następująca (gdzie -n określa numer sygnału):

kill -n PID lub

kill -v nazwa_procesu

Usuwanie procesów **kill** -9 PID

ZADANIA DO WYKONANIA

Zadanie 1. Uruchomić poniższe komendy:

```
cp /dev/zero /dev/null &
```

```
cp /dev/urandom /dev/null &
```

```
cp /dev/zero /dev/null i nacisnąć ^Z (ctrl+Z)
```

```
cp /dev/zero /dev/null i nacisnąć ^Z (ctrl+Z)
```

```
cp /dev/urandom /dev/null i nacisnąć ^Z (ctrl+Z)
```

```
cp /dev/urandom /dev/null i nacisnąć ^Z (ctrl+Z)
```

```
jobs
```

```
jobs -l
```

```
fg %3 i nacisnąć ^Z (ctrl+Z)
```

```
jobs
```

```
kill -9 %2
```

```
jobs
```

Zadanie 2.

- utworzyć 2 dodatkowe konta użytkowników (student1 i student2)
- otworzyć 3 sesje dla administratora (root) (zalogować się na 3 konsolach poprzez np. ctrl+F2).
- zalogować się z wykorzystaniem 2 nowych użytkowników na kolejnych konsolach
- sprawdzić przy pomocy ps pracujące procesy: w tej samej sesji, tylko swoje procesy, wszystkie procesy w systemie.
- przetestować działanie innych opcji ps

Zadanie 3.

- przy pomocy kill usunąć z sesji B procesy pracujące w sesji A
- uruchomić procesy z podwyższoną wartością nice
- zmienić wartość nice pracującego już procesu
- wyświetlić wartość nice procesu

Zadanie 4.

- przeanalizować działanie poniższych komend:

```
ps | sort  
ps -A | grep bash  
ps | head -n 2  
nice du -sh /usr/local/* 2>/dev/null | sort -rn | head
```