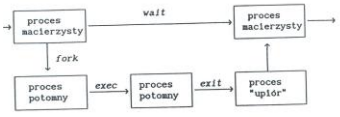
1.



Rys. 6.4. Rozwidlanie procesów

2. 10 zombie

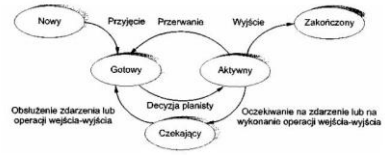
```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
int main()
{
int pid;
for ( int i=0 ; i<10 ; i++)
{
if ( (pid=fork()) == 0 )
{
printf("nr. %d\n", getpid());
sleep(0);
exit(0);
}
else system("ps");
}
return 0;
```

4. max prcesów

```
#!/bin/bash
export RUN=$((RUN + 1))
echo $RUN
$0
```

(ulimit -u <- opcjonalnie)

5. diagram procesów



Rys. 4.1 Diagram stanów procesu

6. skopiować wszystkie pliki dziennika do użytkownika bolek

```
cp  -R  /var/log/  /home/bolek
```

7. czy adres pasuje do podanego gatewaya

Jeżeli dwa urządzenia mają ustawione identyczne wartości maski podsieci, oraz początkowe fragmenty ich adresów IP o długości wyznaczonej przez długość maski są zgodne, wówczas te urządzenia są zaadresowane w tej samej podsieci.

8. head

Print first 10 lines of each FILE to standard output.
With more than one FILE, precede each with a header giving the
file name. With no FILE, or when FILE is -, read standard input.

Options:
-n NUM Print first NUM lines instead of first 10
-c NUM Output the first NUM bytes
-q Never output headers giving file names
-v Always output headers giving file names

9.
tail - odwrotność head

10.
demon Cron

minuta, godzina, miesiac, dzien tygodnia.
* oznacza dowolna wartosc, i oznacza ze bedzie co jedna minute.
mozna tez podawac zakresy
10 0 * * * $HOME/bin/kopia.sh
10 minut po polnocy bedzie codziennie odpalany

raport miesieczny

45 9 10 * * ADRES
kazdego miesiaca o 9 45

0 7 * * 1-5 echio "wstawah" | mail -s "pobudka" bolek
o 7 rano od poniedzialku do piatku wstawaj bolek

proces zombie

jesli proces macierzysty poszedl spac, a proces potomny na niego czeka, jest wtedy zombie ktory dlugo tkwi w systemie

//

W sieci znajduje sie 10 komputerów o numerach IP od 162.123.0.41 do 162.123.0.50 Napisz skrypt który sprawdzi czy komputery są widoczne w sieci (czy jakoś tak);

```
#!/bin/sh
x=40
while [ x -le 49 ]s
do
x=$ [ x + 1 ]
if ping 162.132.0.$x -c1
then echo " 162.123.0.$x jest w siec"
else echo " nie ma 162.123.0.$x w sieci"
fi
done
```

```
#!/bin/sh
ilosc=$(nmap -sP 162.123.0.41-50 | grep "is up" | wc -l)
echo "Ilosc aktywnych komputerow w sieci: " $ilosc
```

Napisz program który stworzy proces potomny i następnie proces potomny wypisze numer procesu potomnego a macierzysty napisze "Wesołych Świąt";

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>

int main()
{
int pid;
if ( (pid=fork()) == 0)
{
printf("nr. %d\n", getpid());
}
else printf("WESOLYCH SWIAT\n");
return 0;
}
```

Napisz jakie zasoby komputera może ograniczyć administrator i w jaki sposób

gdy jest udostepniona usluga SSH, paramet PermitRootLogin trzeba ustawic na no. OBJASNIENIE zeby nie bylo mozna bezposrednio sie zalogowac na konto roota(możliwość zalogowania się na konto roota będzie możliwa tylko po wcześniejszym zalogowaniu się na konto zwykłego użytkownika, po czym będzie istniała możliwość zmiany użytkownika)

Wyjasnic do czego sluzy polecenie sudo

sudo (ang. superuser do) – program stosowany w systemach operacyjnych GNU/Linux, Unix i podobnych, w celu umożliwienia użytkownikom uruchomienia aplikacji, normalnie zarezerwowanych dla administratora zwanego rootem.
Dostęp do tego narzędzia kontroluje zazwyczaj plik /etc/sudoers, w którym wymienieni są wszyscy użytkownicy mogący używać sudo oraz programy, które w ten sposób mogą uruchomić. Edycja tego pliku pozwala kontrolować i zarządzać sudo.
Możliwość korzystania z tego przydatnego, lecz potencjalnie niebezpiecznego narzędzia użytkownik uzyskuje po podaniu swego hasła, co ma utrudnić wykorzystanie programu przez niepowołane osoby w niecnych celach.

Odpowiedź na pytanie o możliwe stany procesu:

a) nowy (new) – w tym stanie proces znajduje się zaraz po utworzeniu;
b) wykonywanie (running, executing) – proces ma przydzielony procesor, który wykonuje jego instrukcje;
c) oczekiwanie (waiting) – proces oczekuje na zdarzenie (np. na zakończenie operacji wejścia-wyjścia);
d) gotowość (ready) – proces oczekuje na przydział procesora w kolejce procesów gotowych do wykonywania;
e) zakończony (terminated) – proces zakończył działanie, lecz wciąż pozostaje w systemie (np. nie przekazał wyników).

Napisz skrypt który skopiuje wszystkie pliki w bieżącym katalogu do pliku o nazwie ./razem , a następnie wyśle ten plik użytkownikowi podanemu jako parametr (czyli $1)

```
#!/bin/bash
for i in ./*; do
cat $i >> ./razem
done
mail $1 < ./razem
exit 0
```