



# OPERÁČNÍ SYSTÉMY II.

C7

2006-11-14

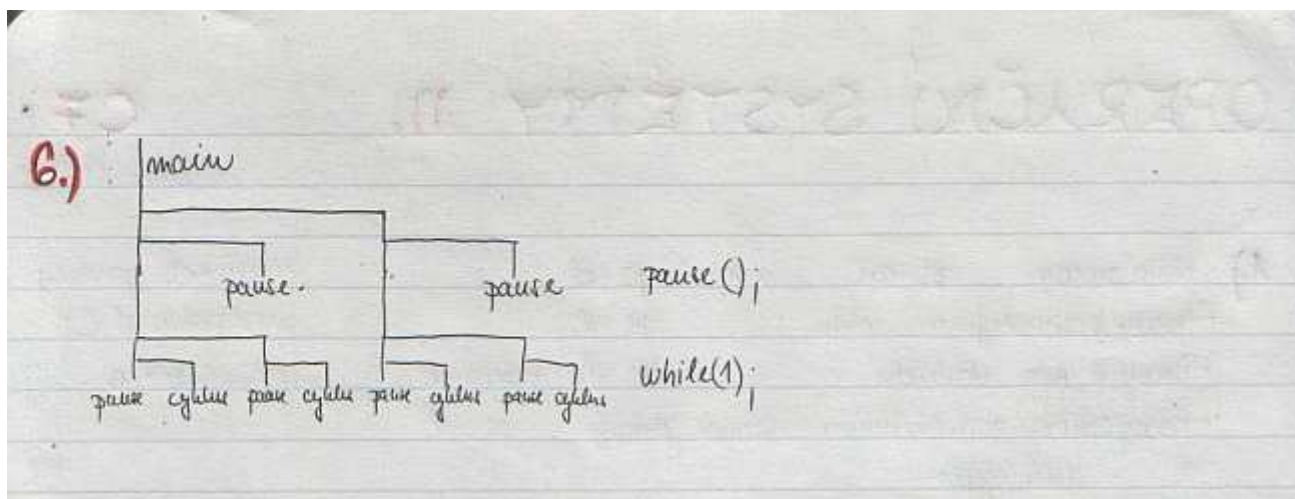
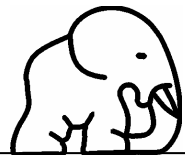
- 1.) • Běžící procesy . `ps aux` nebo `ps -ef` KOMPILE (přeložení)  
• Procesy s přiřazeným terminálem `ps al` PROGRAMU V C:  
• Procesy v mém vlastnictví `ps -fu username` CC název.c  
• Procesy spuštěné z shellu, kterému zadáváme příkazy `ps -l`  
l - delší výpis  
- AND - syntaxe System V  
- NE - syntaxe BSD

- 2.) `main()`  
{  
while(1);  
}  
• Zrušení cyklického programu: `Ctrl+C`  
• Spuštění a zjištění PID rodiče: `ps -ln novak`  
`ps aux | grep "jmeno"`  
PPID - otecův PID  
• Zrušení z daného okna `kill -signal PID`

- 3.) Vypisování na pozadí: `./prog &`

- 4.) • Pozastavit, zablokovat: signál `SIGSTOP (19)` - signál, který způsobí zablokování  
• Odblokovat: signál `SIGCONT (18)` - signál, který způsobí odblokování  
• Zrušit: `kill -signal`

- 5.) Pokud existuje journal, vymaž.  
→ nekonečný cyklus  
Vypis date  
Vypis who  
Vypis \*\*\*  
Sleep 10 sekund  
if [ -f ~/journal ] then rm ~/journal  
fi  
while :  
do  
echo 'date' >> ~/journal  
echo 'who' >> ~/journal  
echo '\*\*\*\*\*' >> ~/journal  
sleep 10  
done  
možná myšlenka  
zdelek z programu



1.
  - a) Vypište všechny procesy, které v systému běží. (`ps axl` nebo `ps -el`)
  - b) Vypište jen procesy s přiřazeným terminálem. (`ps al`)
  - c) Vypište procesy, které máte ve svém vlastnictví. (`ps -lu username`).
  - d) Vypište všechny procesy, které jste spustili z shellu, kterému zadáváte příkazy. (`ps -l`)

Ve všech případech můžete vynechat modifikátor `l` a dostanete zkrácený výpis. Pokud před modifikátory uvádíte `-`, používáte SYSTÉM V syntax. Pokud `-` neuvádíte, používáte BSD syntax.
2. V jazyce C napište nekonečný cyklus. Program spusťte na popředí a sledujte jeho činnost z jiného terminálu.
  - a) Program zrušte v okně, ve kterém běží.
  - b) Program znovu spusťte a zjistěte PID jeho rodiče.
  - c) Sledujte čas spotřebovaný procesem
  - d) Program zrušte z druhého okna zasláním signálu.
3. Program z příkladu 2 spusťte na pozadí a sledujte jeho činnost ze stejného okna. Nakonec program zrušte.
4. Program z příkladu 2 spusťte na pozadí.
  - a) Program pozastavte zasláním signálu a přesvědčte se o jeho zablokování. (Signál, který způsobí pozastavení je `SIGSTOP(19)`).
  - b) Program odblokujte a přesvědčte se, že skutečně běží. (Signál, který způsobí odblokování je `SIGCONT(18)`).
  - c) Nakonec běžící program zrušte.
5. Napište skript, který každých 10sec do souboru `~/journal` vypíše datum a čas a připojené uživatele. Skript vyzkoušejte. Skript pozměňte tak, aby časový interval výpisu bylo možné zadávat parametrem spuštění skriptu.
6. Analýzou programu stanovte kolik procesů v systému poběží a kolik jich zůstane zablokováno, pokud přeložíte a spustíte následující program.

```
main()
{
    fork();
    if(fork{ }==0)
        pause();
    fork();
    if(fork{ }==0)
        while(1);
    pause();
}
```

Výsledek ověřte tak, že program napíšete a spustíte. Nakonec všechny procesy zrušte.