

## Lekcja 7 – Praca domowa

**1. (2 pkt.) Wykorzystując poznane metody napisz program, w którym jeden proces wytwarza 32 procesy potomne. Wykorzystując polecenie pstree, przedstaw drzewo tych procesów**

```
//Przykład z języka C
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/wait.h>

int main()
{
    for(int i=0;i<32;i++)
    {
        if(fork() == 0) //Wywołujemy procesy potomne
        {
            sleep(10);
            exit(0);
        }
    }
    for(int i=0;i<32;i++)
        wait(NULL); //Wstrzymujemy wykonanie procesu macierzystego aż wykonają się potomne
}
```

//Printscreen z polecenia pstree:

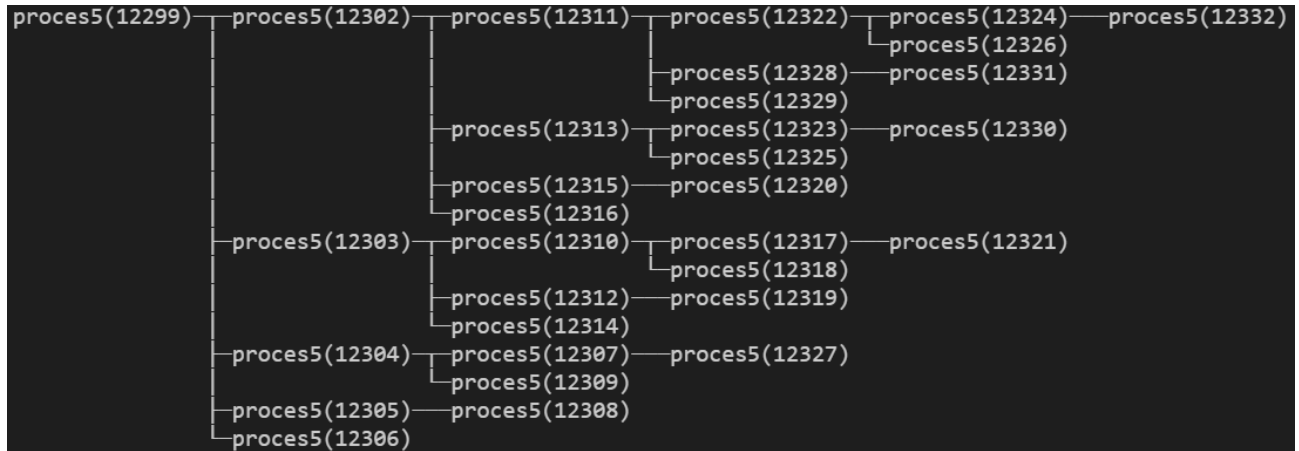
```
proces2(18522)---proces2(18525)
                  |---proces2(18526)
                  |---proces2(18527)
                  |---proces2(18528)
                  |---proces2(18529)
                  |---proces2(18530)
                  |---proces2(18531)
                  |---proces2(18532)
                  |---proces2(18533)
                  |---proces2(18534)
                  |---proces2(18535)
                  |---proces2(18536)
                  |---proces2(18537)
                  |---proces2(18538)
                  |---proces2(18539)
                  |---proces2(18540)
                  |---proces2(18541)
                  |---proces2(18542)
                  |---proces2(18543)
                  |---proces2(18544)
                  |---proces2(18545)
                  |---proces2(18546)
                  |---proces2(18547)
                  |---proces2(18548)
                  |---proces2(18549)
                  |---proces2(18550)
                  |---proces2(18551)
                  |---proces2(18552)
                  |---proces2(18553)
                  |---proces2(18554)
                  |---proces2(18555)
                  |---proces2(18556)
```

**2. (2 pkt.) Wykorzystując poznane metody napisz program, w którym proces macierzysty wytwarza 1 proces potomny, ten proces wytwarza kolejny potomny, kolejny potomny wytwarza kolejny potomny, itd.... tak aby wszystkich potomnych było w sumie 16. Wykorzystując polecenie pstree, przedstaw drzewo tych procesów**

```
//Przykład z języka C
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/wait.h>
int main()
{
    fork();
    fork();
    fork();
    fork();
```

}

```
//Printscreen z polecenia pstree:
```



**3. (2 pkt.) Wyjaśnij różnicę między wątkiem, a procesem.**

Proces (ang. task) w porównaniu z wątkiem jest bardziej złożoną i angażującą większe zasoby strukturą programistyczną. Każdy proces posiada swój osobny segment kodu, segment danych ze zmiennymi globalnymi, stertę do przydzielania pamięci operacyjnej oraz stos do wywoływania procedur i obliczeń. W porównaniu z procesem, wątek jest mniej złożoną i bardziej elastyczną strukturą. Zwykle w ramach jednego procesu współbieżnie wykonuje się kilka wątków, które współdzielą segment kodu, danych i stertę, zaś stos przydzielony jest każdemu z nich indywidualnie do swoich zadań. To współdzielenie części danych przez wątki ułatwia komunikowanie się między nimi, co w przypadku procesów byłoby o wiele trudniejsze i wymagające dodatkowych obliczeń.

4. (2 pkt.) Wykorzystując polecenie systemowe ps wyświetl wszystkie procesy, wraz z informacją o wątkach. Opisz, co jest przedstawione w konkretnej kolumnie twojego listingu. Wykorzystaj polecenie ps.

UID	PID	PPID	LWP	C	NLWP	STIME	TTY	TIME	CMD
root	1	0	1	0	1	23:44	?	00:00:02	/sbin/init
root	2	0	2	0	1	23:44	?	00:00:00	[kthreadd]
root	3	2	3	0	1	23:44	?	00:00:00	[ksoftirqd/0]
root	4	2	4	0	1	23:44	?	00:00:00	[kworker/0:0]
root	5	2	5	0	1	23:44	?	00:00:00	[kworker/0:0H]
root	7	2	7	0	1	23:44	?	00:00:00	[rcu_sched]
root	8	2	8	0	1	23:44	?	00:00:00	[rcuos/0]
root	9	2	9	0	1	23:44	?	00:00:00	[rcuos/1]
root	10	2	10	0	1	23:44	?	00:00:00	[rcuos/2]
root	11	2	11	0	1	23:44	?	00:00:00	[rcuos/3]
root	12	2	12	0	1	23:44	?	00:00:00	[rcuos/4]
root	13	2	13	0	1	23:44	?	00:00:00	[rcuos/5]
root	14	2	14	0	1	23:44	?	00:00:00	[rcuos/6]
root	15	2	15	0	1	23:44	?	00:00:00	[rcuos/7]
root	16	2	16	0	1	23:44	?	00:00:00	[rcuos/8]
root	17	2	17	0	1	23:44	?	00:00:00	[rcuos/9]
root	18	2	18	0	1	23:44	?	00:00:00	[rcuos/10]
root	19	2	19	0	1	23:44	?	00:00:00	[rcuos/11]
root	20	2	20	0	1	23:44	?	00:00:00	[rcuos/12]
root	21	2	21	0	1	23:44	?	00:00:00	[rcuos/13]
root	22	2	22	0	1	23:44	?	00:00:00	[rcuos/14]
root	23	2	23	0	1	23:44	?	00:00:00	[rcuos/15]
root	24	2	24	0	1	23:44	?	00:00:00	[rcuos/16]
root	25	2	25	0	1	23:44	?	00:00:00	[rcuos/17]
root	26	2	26	0	1	23:44	?	00:00:00	[rcuos/18]
root	27	2	27	0	1	23:44	?	00:00:00	[rcuos/19]
root	28	2	28	0	1	23:44	?	00:00:00	[rcuos/20]
root	29	2	29	0	1	23:44	?	00:00:00	[rcuos/21]
root	30	2	30	0	1	23:44	?	00:00:00	[rcuos/22]

W powyższym listingu polecenie `ps -elf` wyświetlone są szczegółowe informacje o procesach i wątkach. W kolumnach mamy informację o user id procesu (UID), numerze procesu (PID, PPID), numerze wątku (LWP). Kolumna TTY podaje informację o terminalu związanym z procesem, TIME podaje skumulowany czas poświęcony przez procesor, CMD podaje informację o nazwie procesu/wątku.

**5. (2 pkt.)** Zaloguj się na komputer [mushelka.pjwstk.edu.pl](http://mushelka.pjwstk.edu.pl) i wykorzystując polecenie systemowe `ps tree` wyświetl drzewo wszystkich procesów z tego komputera. Odnajdź PID procesu swojej powłoki i wskaż go. Czy są tam cztery pierwsze procesy, z drzewa procesów przedstawionych w bloku 4 (rysunek z drzewem procesów systemu Unix)?

```

init(1) -- ModemManager(953) -- {ModemManager}(962)
      | {ModemManager}(969)
      | - NetworkManager(991) -- dhclient(1068)
      |   | dnsmasq(1686)
      |   | {NetworkManager}(999)
      |   | {NetworkManager}(1000)
      |   | {NetworkManager}(1002)
      | - accounts-daemon(1340) -- {accounts-daemon}(1342)
      |   | {accounts-daemon}(1350)
      | - acpid(1311)
      | - avahi-daemon(774) -- avahi-daemon(775)
      | - bluetoothd(747)
      | - colord(2043) -- {colord}(2088)
      |   | {colord}(2086)
      | - cron(1274)
      | - cups-browsed(1351)
      | - cupsd(2210) -- dbus(2214)
      | - dbus-daemon(672)
      | - getty(1196)
      | - getty(1200)
      | - getty(1206)
      | - getty(1207)
      | - getty(1210)
      | - getty(1522)
      | - gnome-keyring-d(2229) -- {gnome-keyring-d}(2230)
      |   | {gnome-keyring-d}(2493)
      |   | {gnome-keyring-d}(2497)
      |   | {gnome-keyring-d}(2514)
      |   | {gnome-keyring-d}(2546)
      |   | {gnome-keyring-d}(2828)
      | - irqbalance(1278)
      | - kerneloops(1261)
      | - lightdm(1308) -- Xorg(1336)
      |   | - lightdm(1590) -- init(2231) -- at-spi-bus-laun(2328) -- dbus-daemon(2348)
      |   |   | {at-spi-bus-laun}(2339)
      |   |   | {at-spi-bus-laun}(2341)
      |   |   | {at-spi-bus-laun}(2349)
      |   |   | - at-spi2-registr(2364) -- {at-spi2-registr}(2385)
      |   |   | - banfdaemon(2326) -- {banfdaemon}(2389)
      |   |   |   | {banfdaemon}(2417)
      |   |   |   | {banfdaemon}(2420)
      |   |   |   | {evolution-calen}(2723)
      |   |   |   | {evolution-calen}(2724)
      |   |   |   | {evolution-calen}(2725)
      |   |   |   | - evolution-sourc(2566) -- {evolution-sourc}(2618)
      |   |   |   |   | {evolution-sourc}(2634)
      |   |   |   | - gconfd-2(2697)
      |   |   |   | - gnome-session(2380) -- compiz(2601) -- {compiz}(2607)
      |   |   |   |   | {compiz}(2612)
      |   |   |   |   | {compiz}(2617)
      |   |   |   |   | {compiz}(3066)
      |   |   |   |   | {compiz}(3067)
      |   |   |   | - deja-dup-monito(2991) -- {deja-dup-monito}(2994)
      |   |   |   |   | {deja-dup-monito}(2995)
      |   |   |   | - nautilus(2622) -- {nautilus}(2657)
      |   |   |   |   | {nautilus}(2679)
      |   |   |   |   | {nautilus}(2728)
      |   |   |   | - nm-applet(2633) -- {nm-applet}(2669)
      |   |   |   |   | {nm-applet}(2675)
      |   |   |   | - polkit-gnome-au(2625) -- {polkit-gnome-au}(2651)
      |   |   |   |   | {polkit-gnome-au}(2662)
      |   |   |   | - telepathy-indic(2767) -- {telepathy-indic}(2772)
      |   |   |   |   | {telepathy-indic}(2773)
      |   |   |   | - unity-fallback-(2619) -- {unity-fallback-}(2639)
      |   |   |   |   | {unity-fallback-}(2684)
      |   |   |   | - update-notifier(2941) -- {update-notifier}(2944)
      |   |   |   |   | {update-notifier}(2945)
      |   |   |   |   | {update-notifier}(2946)
      |   |   |   | - zeitgeist-datab(2788) -- {zeitgeist-datab}(2791)
      |   |   |   |   | {zeitgeist-datab}(2802)
      |   |   |   |   | {zeitgeist-datab}(2803)
      |   |   |   |   | {zeitgeist-datab}(2806)
      |   |   |   | - {gnome-session}(2448)
      |   |   |   | - {gnome-session}(2450)
      |   |   |   | - {gnome-session}(2456)
      |   |   |   | - gnome-terminal(2872) -- bash(2882) -- less(3211)
      |   |   |   |   | pstree(3210)
      |   |   |   |   | - gnome-pty-helper(2881)
      |   |   |   |   | {gnome-terminal}(2875)
      |   |   |   |   | {gnome-terminal}(2876)

```

Powyżej przedstawiono częściowy zrzut ekranu z polecenia `pstree -p | less`, wyświetlającego drzewko wszystkich procesów wraz z numerami PID. Użyto komendy `less`, by łatwiej było przeglądać całe drzewko procesów. Poniżej zaznaczono w obramowaniu PID mojej powłoki `bash` (2882). Na tym drzewku widać jedynie pierwszy (`init`) z czterech procesów drzewa procesów systemu Unix.

Przygotował: Jakub Woźniak [4135@pja.edu.pl](mailto:4135@pja.edu.pl)