Zadanie 3

Ustalenie notacji

Proces Q jest **przodkiem** procesu P jeśli $P \neq Q$ oraz istnieje ciąg procesów $P = P_1, P_2, \ldots P_n= Q$, taki że P_{i+1} jest rodzicem procesu P_i dla każdego $i = 1, 2, \ldots, n-1$. W szczególności relacja bycia przodkiem jest przeciwzwrotna i przechodnia.

Proces Q jest **najniższym wspólnym przodkiem** procesów P_1 i P_2, jeśli:

- Q jest przodkiem zarówno P_1, jak i P_2;
- każdy proces R różny od Q, który jest przodkiem zarówno P_1, jak i P_2, jest również przodkiem Q.

Uwaga: Czasami w literaturze pojawiają się inne definicje najniższego wspólnego przodka.

Uwaga2: Poprawka w definicji najniższego wspólnego przodka na czerwono.

Nowe wywołanie systemowe

Zadanie polega na dodaniu nowego wywołania systemowego `PM_GETLCAPID` z funkcją biblioteczną `pid_t getlcapid(pid_t pid_1, pid_t pid_2)` zadeklarowaną w pliku `unistd.h`.

Wywołanie systemowe `PM_GETLCAPID` przekazuje w wyniku identyfikator procesu, który jest najniższym wspólnym przodkiem dwóch zadanych procesów.

Funkcja `pid_t getlcapid(pid_t pid_1, pid_t pid_2)` przekazuje w wyniku identyfikator najniższego wspólnego przodka procesów o identyfikatorach `pid_1` i `pid_2`.

Jeśli któryś z procesów o identyfikatorach `pid_1` lub `pid_2` nie jest aktualnie działającym procesem, funkcja przekazuje w wyniku `-1` i ustawia `errno` na `EINVAL`.

Jeśli dla danych procesów o identyfikatorach `pid_1` i `pid_2` nie istnieje dokładnie jeden najniższy wspólny przodek, funkcja przekazuje w wyniku `-1` i ustawia `errno` na `ESRCH`.

Format rozwiązania

Poniżej przyjmujemy, że `ab123456` oznacza identyfikator studenta rozwiązującego zadanie. Należy przygotować łatkę (ang. *patch*) ze zmianami w katalogu `/usr`. Plik zawierający łatkę o nazwie `ab123456.patch` uzyskujemy za pomocą polecenia

```
diff -rupNEZbB oryginalne-źródła/usr/ moje-rozwiązanie/usr/ > ab123456.patch
```

gdzie `oryginalne-źródła` to ścieżka do niezmienionych źródeł MINIX-a, natomiast `moje-rozwiązanie` to ścieżka do źródeł MINIX-a zawierających rozwiązanie. Tak użyte polecenie `diff` rekurencyjnie przeskanuje pliki ze ścieżki `oryginalne-źródła/usr`, porówna je z plikami ze ścieżki `moje-rozwiązanie/usr` i wygeneruje plik `ab123456.patch`, który podsumowuje różnice. Tego pliku będziemy używać, aby automatycznie nanieść zmiany na czystą kopię MINIX-a, gdzie będą przeprowadzane testy rozwiązania. Więcej o poleceniu `diff` można dowiedzieć się z podręcznika (`man diff`).

Umieszczenie łatki w katalogu `/` na czystej kopii MINIX-a i wykonanie polecenia `patch -p1 < ab123456.patch` powinno skutkować naniesieniem wszystkich oczekiwanych zmian wymaganych przez rozwiązanie. Należy zadbać, aby łatka zawierała tylko niezbędne różnice.

Po naniesieniu łatki zostaną wykonane polecenia:

- `make && make install` w katalogach `/usr/src/minix/servers/pm` oraz `/usr/src/lib/libc`,
- `make do-hdboot` w katalogu `/usr/src/releasetools`,
- `reboot`.

Rozwiązanie w postaci łatki `ab123456.patch` należy umieścić na Moodlu.

Uwagi

- Serwer PM przechowuje informacje o procesach w tablicy `mproc` zadeklarowanej w pliku `mproc.h`. Działające procesy mają ustawioną flagę `IN_USE`.
- Warto przeanalizować jak PM realizuje wywołania systemowe. Więcej informacji o działaniu tego serwera będzie na laboratorium 8.
- Należy samodzielnie przetestować rozwiązanie. Przykładowy scenariusz wygląda następująco. Załóżmy, że użytkownik stworzył proces o identyfikatorze `187`, a ten proces w wyniku funkcji `fork` stworzył trzy procesy o identyfikatorach `188`, `189` i `190`. Następnie proces o identyfikatorze `188` stworzył proces o identyfikatorze `191`. Wtedy wywołanie `getlcapid(190, 191)` powinno zwrócić `187`.
- Nie przyznajemy punktów za rozwiązanie, w którym łatka nie nakłada się poprawnie, które nie kompiluje się lub które powoduje **kernel panic** podczas uruchamiania systemu.