

Zadanie - zestaw 3

Otwarto: wtorek, 24 marca 2015, 01:00

Tworzenie procesów. Środowisko procesu, sterowanie procesami.

Zadanie 1. (20%)

Zmodyfikuj zadanie 2 z poprzedniego zestawu w taki sposób, iż przeszukiwanie w każdym z odnalezionych (pod)katalogów powinno odbywać się w osobnym procesie.

Zadanie 2. Prosty interpreter zadań wsadowych (40%)

W ramach ćwiczenia należy napisać prosty interpreter zadań wsadowych. Interpreter przyjmuje w argumentach nazwę pliku zawierającego zadanie wsadowe i wykonuje wszystkie polecenia z tego pliku. Polecenia w pliku wsadowym mają postać:

```
NazwaProgramu arg1 arg2 ...
```

co oznacza, że należy wykonać program o nazwie `NazwaProgramu` z argumentami: `arg1`, `arg2`, ...

Na przykład, linia postaci:

```
ls -l
```

powinna spowodować wykonanie programu `ls` z argumentem `-l`. Lista argumentów może być pusta - wówczas program wykonywany jest bez argumentów. W zadaniu można również przyjąć górne ograniczenie na liczbę argumentów.

Interpreter musi wykonywać polecenia w osobnych procesach. W tym celu, po odczytaniu polecenia do wykonania interpreter tworzy nowy proces potomny. Proces potomny natychmiast wykonuje odpowiednią funkcję z rodziny `exec`, która spowoduje uruchomienie wskazanego programu z odpowiednimi argumentami. Uwaga: proces potomny powinien uwzględniać zawartość zmiennej środowiskowej `PATH` - polecenie do wykonania nie musi obejmować ścieżki do uruchamianego programu, jeśli program ten znajduje się w katalogu wymienionym w zmiennej `PATH`.

Po stworzeniu procesu potomnego, proces interpretera czeka na jego zakończenie i odczytuje status zakończenia. Jeśli proces zakończył się ze statusem 0 interpreter przystępuje do wykonania kolejnej linii pliku wsadowego. W przeciwnym wypadku interpreter wyświetla komunikat o błędzie i kończy pracę. Komunikat ten powinien wskazywać, które polecenie z pliku wsadowego zakończyło się błędem. Zakładamy, że polecenia z pliku wsadowego nie oczekują na żadne wejście z klawiatury. Mogą natomiast wypisywać wyjście na ekran.

Zadanie 3. Zasoby procesów

a) (30%)

Zmodyfikuj program z Zadania 2 tak, aby każde polecenie wykonywane przez interpreter miało nałożone pewne twarde ograniczenie na dostępny czas procesora oraz rozmiar pamięci wirtualnej. Wartości tych ograniczeń (odpowiednio w sekundach i megabajtach) powinny być przekazywane jako drugi i trzeci argument wywołania interpretera (pierwszym argumentem jest nazwa pliku wsadowego). Ograniczenia powinny być nakładane przez proces potomny, bezpośrednio przed wywołaniem funkcji z rodziny `exec`. W tym celu proces potomny powinien użyć funkcji `setrlimit`. Zakładamy, że wartości nakładanych ograniczeń są niższe (t.j. bardziej restrykcyjne) niż ograniczenia, które system operacyjny narzuca na użytkownika uruchamiającego interpreter.

Zaimplementuj w interpreterze raportowanie zużycia zasobów systemowych dla każdego wykonywanego polecenia (a więc linii pliku wsadowego). Interpreter powinien w szczególności raportować czas użytkownika i czas systemowy. Realizując tą część zadania zwróć uwagę na funkcję `getrusage` i flagę `RUSAGE_CHILDREN`.

b) 10%

Na potrzeby demonstracji zadania napisz prosty program, który przekracza narzucone limity na zasoby systemowe. Program może np. wykonywać pętlę nieskończoną lub alokować (i zapisywać) znaczną ilość pamięci operacyjnej.