Lista 1

Termin wykonania: 2021-04-11

(Na liście jest jedno zadanie, ale za rozwiązanie w każdym z dwóch języków Ada i Go są osobne punkty do zdobycia.)

Zadanie 1.

Punktacja:

- rozwiązanie w Go: maksymalnie 3 punkty
- rozwiązanie w Adzie: maksymalnie 2.5 punktu
- Można zrobić obydwa rozwiązania.
- Oprócz poprawności implementacji znaczenie ma jakość prezentacji.

Zaimplementować program współbieżny symulujący system, w którym nadawca wysyła do odbiorcy autonomiczne pakiety podróżujące po acyklicznym grafie skierowanym G z jednym źródłem i z jednym ujściem.
Wierzchołki grafu G są indeksowane liczbami naturalnymi od 0 do $n-1$.
Wierzchołek 0 jest źródłem, a wierzchołek $n-1$ jest ujściem.
Dla każdego i, $0 \leq i \leq n-2$, istnieje w grafie krawędź skierowana $(i,\ i+1)$, co zapewnia, że z każdego wierzchołka jest jakaś ścieżka skierowana do ujścia.
Ponadto w grafie istnieje pewna liczba d dodatkowych krawędzi postaci (j,k), gdzie $0 \leq j < k \leq n-1$ (tzw. *skrótów*).
Dla wierzchołka i, *zbiorem następników* i, $N(i)$, nazywamy zbiór takich j, że istnieje krawędź (i,j).
Program ma działać następująco:

1. Generowany jest graf G dla podanych parametrów n i d, gdzie d *skrótów* generowane jest w sposób losowy.
2. Graf G drukowany jest na terminalu tak aby przedstawić istniejące połączenia. (Zastanowić się nad tym jaki sposób prezentacji będzie najbardziej czytelny.)
3. Uruchamiana jest symulacja systemu przesyłania pakietów po grafie G.

System przesyłania pakietów działa według następujących zasad:

- W jednym wierzchołku może przebywać tylko jeden pakiet.
- Co pewien losowy czas nadawca umieszcza w źródle (o ile jest ono puste) nowy pakiet indeksowany kolejną liczbą naturalną.
- Co pewien losowy czas odbiorca odbiera z ujścia pakiet (o ile jest co odebrać).
- Pakiet w wierzchołku i, po odczekaniu losowego czasu, wybiera losowo jeden wierzchołek j ze zbioru $N(i)$ i czeka aż będzie mógł się do niego przemieścić.
- Gdy pakiet p dotrze do wierzchołka i drukowany jest komunikat:
  "pakiet p jest w wierzchołku i"
  i jednocześnie p dodaje i do swojej listy odwiedzonych wierzchołków oraz i dodaje p do swojej listy obsłużonych pakietów.
- Gdy odbiorca odbierze pakiet p, drukuje komunikat:
  "pakiet p został odebrany".
- Po nadaniu $k$pakietów, nadawca kończy nadawanie.

Gdy odbiorca odbierze ostatni (tj. k-ty) pakiet, system kończy działanie i rozpoczyna się drukowanie raportów końcowych.
W raportach końcowych pojawią się dwa wykazy:

- dla każdego wierzchołka, lista kolejno obsłużonych przez niego pakietów,
- dla każdego pakietu, lista odwiedzonych przez niego wierzchołków (ścieżka od źródła do ujścia).

Zaimplementuj system tak, aby nadawca, odbiorca oraz każdy wierzchołek grafu były osobnymi wątkami współbieżnymi, które przekazują sobie pakiety przez narzędzia komunikacji między wątkami.
Zwróć uwagę, że komunikaty drukowane przez współbieżne wątki na terminalu mogą się przeplatać. Dlatego dodaj nowy wątek (serwer drukowania), który przyjmuje zlecenia drukowania komunikatów od wątków i niezwłocznie, ale sekwencyjnie, drukuje je na ekranie w osobnych liniach.

Przetestuj działanie swojego programu przy różnych wartościach parametrów n, d, k i różnych ograniczeniach na losowe opóźnienia w działaniach wątków.
Do prezentacji w `asciinema` wybierz takie parametry aby jak najlepiej to wyglądało.

Wskazówki:

- Każdy koniec kanału w Go może być używany przez różne współbieżne wątki.
- Obejrzyj podgląd pliku `common_channel.go (w tym samym katalogu Dysku Google) zawierający` przykład wspólnego używania kanału przez różne wątki oraz wykorzystania generatora liczb losowych.