

Temat: System rezerwacji zasobów

Opis:

System obliczeniowy składa się ze zbioru procesów rozmieszczonych w sieci. Każdy z procesów posiada pewien zbiór różnych zasobów. Procesy tworzą sieć logiczną, której organizacja zależy od implementacji a jej węzłami są poszczególne procesy. Sieć jest tworzona w sposób przyrostowy poprzez uruchamianie kolejnych procesów (tworzenie kolejnych węzłów) i podłączanie ich do już istniejącej sieci. Każdy proces po starcie podpiną się do sieci zgodnie z założonym modelem komunikacji w sieci i oczekuje na ewentualne połączenia od kolejnych nowych składowych systemu (kolejnych dodawanych węzłów) lub klientów pragnących zarezerwować zasoby. Komunikacja z potencjalnymi klientami oraz kolejnymi nowo podłączanymi węzłami odbywa się za pomocą protokołu TCP. Komunikacja z węzłami tworzącymi sieć w trakcie alokacji zasobów może odbywać się zarówno za pomocą TCP jak i UDP – wybór pozostaje w gestii realizatora zadania.

Uruchomienie węzła sieci ma następującą postać:

```
java NetworkNode -ident <identyfikator> -tcpport <numer portu TCP>
-gateway <adres>:<port> <lista zasobów>
```

gdzie:

- `-ident <identyfikator>` określa identyfikator danego węzła (liczba naturalna, unikalna w obrębie sieci).
- `-tcpport <numer portu TCP>` określa numer portu TCP na którym dany węzeł sieci oczekuje na połączenia od klientów.
- `-gateway <adres>:<port>` oznacza adres IP oraz numer portu TCP, na którym oczekuje jeden z już uruchomionych węzłów sieci. Dla pierwszego węzła sieci parametru tego nie podaje się.
- `<lista zasobów>` oznacza niepustą listę zasobów jakimi dysponuje dany węzeł w postaci par: `<typ zasobu>:<liczba>`, gdzie `<typ zasobu>` to jednoliterowy identyfikator typu zasobu a `<liczba>` określa liczbę sztuk tego zasobu przypisanych do danego węzła.

Przykład wywołania:

```
java NetworkNode -ident 123 -tcpport 9991 -gateway localhost:9990 A:5 C:3
```

Oznacza to uruchomienie węzła o identyfikatorze 123, który nasłuchuje na połączenia od klientów lub innych nowych węzłów na porcie 9991 a do podłączenia siebie do sieci wykorzysta węzeł pracujący na komputerze o adresie `localhost` i porcie TCP o numerze 9990. Do procesu tego jest przydzielonych początkowo 5 sztuk zasobów typu A oraz 3 sztuki zasobów typu C.

Po uruchomieniu sieci, można do niej podpinąć klientów. Klient jest prostym procesem, który ma za zadanie wysłać do sieci żądanie rezerwacji zasobów w odpowiednim formacie i poczekać na odpowiedź. Podczas uruchomienia klient otrzymuje jako parametr adres IP jednego z węzłów sieci (dowolnego, zwanego dalej „węzłem kontaktowym”), numer portu TCP na którym ten węzeł nasłuchuje na zgłoszenia od klientów oraz listę zasobów do alokacji (w takim samym formacie jak opisany powyżej dla węzłów sieci). Przykładowe wywołanie klienta:

```
java NetworkClient -ident 321 -gateway localhost:9991 A:3 C:2 B:1
```

oznacza uruchomienie klienta o identyfikatorze 321, który do połączenia z siecią ma wykorzystać węzeł działający na komputerze `localhost` i porcie TCP o numerze 9991. Po połączeniu ma podjąć próbę alokacji 3 sztuk zasobu A, 1 sztuki zasobu B oraz 2 sztuk zasobu C.

Po uzyskaniu połączenia klient przesyła w formie pojedynczej linii tekstu komunikat o następującym formacie:

```
<identyfikator> <zasób>:<liczność> [<zasób>:liczność]
```

czyli swój identyfikator będący liczbą naturalną (dowolną) po którym następuje niepusta sekwencja par postaci nazwa zasobu:liczba sztuk do rezerwacji rozdzielonych pojedynczą spacją. Węzeł systemu o odebraniu takiego komunikatu podejmuje próbę zaalokowania w systemie zasobów zgodnie z żądaniem klienta. Realizacja tej alokacji oraz schemat komunikacji pozostaje w gestii realizatora zadania. Zasoby mogą zostać zaalokowane na dowolnych węzłach sieci, z dowolnym ich rozkładem (np. jeśli klient żąda 3 sztuk zasobu A to jedna jego sztuka może zostać zaalokowana na węzeł 1 a pozostałe dwie na węzeł 2) – o ile tylko jest to wykonalne (cały system posiada łącznie odpowiednią liczbę zasobów). Alokacja polega na zapamiętaniu w każdym węźle uczestniczącym w transakcji jakie zasoby i w jakiej liczbie są przydzielone do danego klienta (według identyfikatora) oraz adekwatnym zmniejszeniu liczby dostępnych w danym węźle zasobów. W danym węźle można przydzielić klientowi nie więcej sztuk danego zasobu jak liczba aktualnie dostępnych. Można natomiast przydzielić zasób w różnych węzłach sieci tak, żeby łącznie uzyskać wymaganą przez klienta ilość.

Alokacja może się powieść – lecz nie musi. Jeśli odbędzie się poprawnie, po zebraniu informacji zwrotnej od węzłów sieci o tym, kto co i w jakiej liczbie zdołał zaalokować, węzeł kontaktowy odsyła do klienta raport rozpoczynający się od wiersza następującej postaci:

```
ALLOCATED
```

po czym następuje seria wierszy wierszy postaci:

```
<zasób>:<liczność>:<ip węzła>:<port węzła>
```

po czym kończy połączenie. Jeśli alokacja się nie powiedzie z powodu braku dostatecznej liczba jakiegokolwiek z żądanych zasobów (system musi zaalokować albo wszystko co klient żąda albo nic), węzeł kontaktowy wysyła do klienta komunikat zawierający jeden wiersz:

```
FAILED
```

po czym kończy połączenie z klientem.

W obu przypadkach klient oczekuje na odpowiedzi, które wyświetla na konsoli a po zakończeniu połączenia kończy działanie.

Dodatkowa funkcjonalność pozwala na zakończenie pracy systemu. Zamiast komunikatu opisanego powyżej węzeł sieci może otrzymać komunikat zawierający jeden wiersz postaci:

```
TERMINATE
```

W takim przypadku węzeł ten kończy połączenie z klientem i inicjuje zakończenie działania sieci, w wyniku którego wszystkie węzły sieci powinny zakończyć pracę.

Uwagi:

- Zasoby są między sobą niewymienne, czyli zasób typu A nie może zostać zamieniony na zasób typu B i odwrotnie.
- Na danym fizycznym komputerze może być uruchomionych wiele węzłów sieci logicznej. Należy więc odpowiednio dbać o wykorzystanie portów aby uniknąć konfliktów.
- Jeśli alokacja odbędzie się poprawnie, węzły w niej uczestniczące mogą wypisać na konsoli komunikat o tym informujący, np. z liczbą pozostałych zasobów.

- Jeśli żądanie przydziału zasobów nie może zostać spełnione w całości, po zakończeniu próby jego wykonania system ma znajdować się w takim stanie jak przed próbą.
- Należy koniecznie przestrzegać zarówno nazewnictwa głównej klasy (jak w przykładach powyżej) jak i sposobu uruchamiania aplikacji – do testów zostanie użyty zautomatyzowany system nietolerujący odchyłań.
- Nie trzeba pisać procesu klienta – dostarczymy go Państwu w formie gotowej na początku stycznia.

Specyfikacja:

- 1 Zadanie polega na zaprojektowaniu i zaimplementowaniu protokołu oraz aplikacji go wykorzystującej, która umożliwi organizację sieci oraz późniejszą rezerwację zasobów przez procesy klienckie – zgodnie z opisem.
 - 1.1 Określenie sposobu organizacji sieci węzłów, podłączania nowych do już istniejącej sieci, oraz metody ich komunikacji w celu realizacji alokacji jest elementem zadania.
 - 1.2 Wybór węzłów na których odbędzie się alokacja poszczególnych zasobów (w miarę ich dostępności) również jest elementem zadania.
- 2 Projekty powinny zostać zapisane do odpowiednich katalogów (w sekcji Zadania) w systemie GAKKO do 23.01.2022 (termin może zostać zmieniony przez prowadzącego daną grupę).
- 3 Za poprawne rozwiązanie tego zadania można uzyskać do 500 punktów:
 - 3.1 Maksymalnie 200 punktów za kompletną dokumentację opisującą sposób organizacji sieci, opis komunikacji wraz z opisem przesyłanych komunikatów i protokołu komunikacji. **Obecność poprawnej dokumentacji jest warunkiem koniecznym do tego, aby zostały sprawdzone pozostałe elementy projektu** (3.2, 3.3 i 3.4).
 - 3.2 Maksymalnie 100 punktów za implementację procesu węzła sieci umożliwiającej organizację takiej sieci (podłączanie kolejnych węzłów oraz terminację całości).
 - 3.3 Maksymalnie kolejne 100 punktów za uzupełnienie procesu o funkcjonalności pozwalające na odbieranie żądań od klienta i wykonanie rezerwacji przy założeniu, że wybrany węzeł kontaktowy posiada wolne zasoby w wymaganej liczności (konieczna implementacja funkcjonalności punktu 3.2).
 - 3.4 Maksymalnie kolejne 100 punktów za implementację systemu umożliwiającą dowolną alokację zasobów (konieczna implementacja funkcjonalności punktów 3.2 i 3.3).
- 4 Spakowane archiwum z plikami projektu musi zawierać:
 - 4.1 Pliki źródłowe (dla JDK 1.8),
 - 4.2 Pliki binarne (class),
 - 4.3 Skrypty startowe, umożliwiające uruchomienie implementacji (opcjonalne),
 - 4.4 Plik Readme.txt z opisem i uwagami autora, w szczególności:
 - 4.4.1 **szczegółowy opis implementacji**, w szczególności szczegółowy opis protokołu (opis budowy komunikatów i ich przetwarzania, wymagane dla punktu 3.4).
 - 4.4.2 jak skompilować i zainstalować (z linii poleceń bez używania GUI),
 - 4.4.3 co zostało zaimplementowane,
 - 4.4.4 co nie działa (jeśli nie działa).
- 5 JEŚLI NIE WYSZCZEGÓLNIŁO INACZEJ, WSZYSTKIE NIEJASNOŚCI NALEŻY PRZEDYSKUTOWAĆ Z PROWADZĄCYM ZAJĘCIA POD GROŹBĄ NIEZALICZENIA PROGRAMU W PRZYPADKU ICH NIEWŁĄŚCIWEJ INTERPRETACJI.