

Примерни задачи за курс “Операционни системи”, СУ, ФМИ

28 февруари 2020 г.

Семинарните упражнения на курса “Операционни системи” разглеждат следните теми, групирани по “Работа в GNU/Linux shell” и “Използване на системни примитиви в програми на C”:

1. Въведение (shell)
2. Файлова система и работа с файлове (shell)
3. Обработка на текст (shell)
4. Процеси (shell)
5. Командни интерпретатори и скриптове (shell)
6. Системни примитиви за вход и изход (C)
7. Системни примитиви за работа с процеси (C)
8. Системни примитиви за работа с pipe-ове (C)

Тук са събрани примерни задачи по различните теми, както и някои теоретични въпроси, с надеждата те да бъдат полезни на студентите при тяхната работа в курса. Очаква се студентите да са разгледали някои по-базови задачи по дадена тема преди да преминат към изложените тук.

Работа в GNU/Linux shell

Забележка: За всички задачи, освен ако не е указано друго, в имената на файловете и директориите няма специални символи. Във файловата система може да съществуват директории, до които нямате достъп.

Задачи за теми 1,2,3

Зад. 1 Даден е текстов файл с име `philip-j-fry.txt`. Напишете shell script и/или серия от команди, които извеждат броя редове, съдържащи поне една четна цифра и несъдържащи малка латинска буква от `a` до `w`.

Примерно съдържание на файла:

```
123abv123
123zz123
MMU_2.4
```

Примерен изход:

Броят на търсените редове е 2

Зад. 2 Имате текстов файл със следното съдържание (всяка книга е на един ред):

```
1979 г. - „Синият тайфун“ (сборник съветски научнофантастични разкази за морето)
1979 г. - „Двойната звезда“ - Любен Дилов
1979 г. - „Завръщане от звездите“ - Станислав Лем (Превод: Веселин Маринов)
1979 г. - „Среща с Рама“ - Артър Кларк (Превод: Александър Бояджиев)
1979 г. - „Алиби“ - Димитър Пеев (криминален роман)
1979 г. - „Тайнственият триъгълник“ (сборник НФ разкази за морето)
1979 г. - „Второто нашествие на марсианците“ - Аркадий и Борис Стругацки
```

1979 г. - „Гробищен свят“ - Клифърд Саймък (Превод: Михаил Грънчаров)
 1979 г. - „Чоки“ - Джон Уиндъм (Превод: Теодора Давидова)
 1979 г. - „Спускане в Маелстрьом“ - Едгар Алан По (Превод: Александър Бояджиев)
 1980 г. - „Допълнителна примамка“ - Робърт Ф. Йънг (Превод: Искра Иванова, ...)
 1980 г. - „Кристалното яйце“ - Хърбърт Уелс (Превод: Борис Миндов, ...)
 1980 г. - „Онирофилм“ (сборник италиански НФ разкази) (Превод: Никола Иванов, ...)

Напишете shell script (приемащ аргумент име на файл) и серия от команди, които извеждат:

- всеки ред от файла с добавен пореден номер във формат "1. ", "2. ", ... "11. " ...
- махат данните за годината на издаване
- сортират изхода по заглавие (лексикографски, възходящо)

Примерен изход (показани са само първите 4 реда):

5. „Алиби“ - Димитър Пеев (криминален роман)
 7. „Второто нашествие на марсианците“ - Аркадий и Борис Стругацки
 8. „Гробищен свят“ - Клифърд Саймък (Превод: Михаил Грънчаров)
 2. „Двойната звезда“ - Любен Дилов

Зад. 3 В текущата директория има само обикновени файлове (без директории). Да се напише bash script, който приема 2 позиционни параметъра – числа, които мести файловете от текущата директория към нови директории (a, b и c, които трябва да бъдат създадени), като определен файл се мести към директория 'a', само ако той има по-малко редове от първи позиционен параметър, мести към директория 'b', ако редове са между първи и втори позиционен параметър и в 'c' в останалите случаи.

Зад. 4 Файловете във вашата home директория съдържат информация за музикални албуми и имат специфична структура. Началото на всеки ред е годината на издаване на албума, а непосредствено, след началото на всеки ред следва името на изпълнителя на песента. Имената на файловете се състоят от една дума, която съвпада с името на изпълнителя.

Примерно съдържание на файл с име "Bonnie":

2005г. Bonnie - "God Was in the Water" (Randall Bramblett, Davis Causey) - 5:17
 2005г. Bonnie - "Love on One Condition" (Jon Cleary) - 3:43
 2005г. Bonnie - "So Close" (Tony Arata, George Marinelli, Pete Wasner) - 3:22
 2005г. Bonnie - "Trinkets" (Emory Joseph) - 5:02
 2005г. Bonnie - "Crooked Crown" (David Batteau, Maia Sharp) - 3:49
 2005г. Bonnie - "Unnecessarily Mercenary" (Jon Cleary) - 3:51
 2005г. Bonnie - "I Will Not Be Broken" - "Deep Water" (John Capek, Marc Jordan) - 3:58

Да се състави процедура на bash приемаща два параметъра, които са имена на файлове от вашата home директория. Скриптът сравнява, кой от двата файла има повече на брой редове, съдържащи неговото име (на файла). За файлът победител изпълнете следните действия:

- извлекете съдържанието му, без годината на издаване на албума и без името на изпълнителя
- сортирайте лексикографски извлеченото съдържание и го запишете във файл с име 'изпълнител.songs'

Примерен изходен файл (с име Bonnie.songs):

"Crooked Crown" (David Batteau, Maia Sharp) - 3:49
 "God Was in the Water" (Randall Bramblett, Davis Causey) - 5:17
 "I Will Not Be Broken" - "Deep Water" (John Capek, Marc Jordan) - 3:58
 "Love on One Condition" (Jon Cleary) - 3:43
 "So Close" (Tony Arata, George Marinelli, Pete Wasner) - 3:22
 "Trinkets" (Emory Joseph) - 5:02
 "Unnecessarily Mercenary" (Jon Cleary) - 3:51

Зад. 5 Напишете серия от команди, извеждащи на екрана само броя на всички обекти във файловата система, чиито собственик е текущият потребител.

Забележка: Във файловата система със сигурност съществуват директории, до които нямате достъп.

Зад. 6 Напишете серия от команди, които изтриват:

- а) всички файлове в текущата директория и нейните поддиректории, които са с нулева дължина.
- б) 5-е най-големи файла в home директорията на текущия потребител и нейните поддиректории.

Зад. 7 Напишете серия от команди, които от файла `/etc/passwd` да вземат под-низ, състоящ се от втора и трета цифра на факултетния номер на студентите от специалност Информатика, чиито фамилии завършват на "а". Изведете коя комбинация от цифри се среща най-често и коя е тя.

Примерно съдържание на файла:

```
s45194:x:1255:502:Elizabet Mihaylova, Inf, k3, g1:/home/Inf/s45194:/bin/bash
s45139:x:1261:502:Vasilena Peycheva:/home/Inf/s45139:/bin/bash
s81257:x:1079:503:Vasilena Nikolova, KN, 2kurs, 5gr:/home/KN/s81257:/bin/bash
s81374:x:1117:503:Ivan Kamburov, KN, 2kurs, 7gr:/home/KN/s81374:/bin/bash
kiril:x:508:500:Kiril Varadinov:/home/kiril:/bin/bash
s61812:x:1128:504:Vladimir Genchev:/home/SI/s61812:/bin/bash
user:x:1000:99:Inactive user just to start UID from 1000:/home/user:/sbin/nologin
s81254:x:1077:503:Mariela Tihova, KN, 2kurs, 5gr:/home/KN/s81254:/bin/bash
s81386:x:1121:503:Daniela Ruseva, KN, 2kurs, 7gr:/home/KN/s81386:/bin/bash
s45216:x:1235:502:Aleksandar Yavashev, Inf, k3, g3:/home/Inf/s45216:/bin/bash
```

Примерен изход:

2 51

Зад. 8 Намерете имената на топ 5 файловете в текущата директория с най-много hardlinks.

Зад. 9 Напишете серия от команди, извеждащи на екрана *само* inode-а на най-скоро променения (по съдържание) файл, намиращ се в home директорията на потребител resho (или нейните под-директории), който има повече от едно име.

Зад. 10 При подреждане в нарастващ ред на числовите потребителски идентификатори (UID) на акаунтите, дефинирани в системата, 201-ят акаунт е от групата, запазена за акаунти от специалност СИ.

Изведете списък с имената (име и фамилия) и home директориите на всички акаунти от специалност СИ, подреден по факултетен номер.

За справка:

```
s61988:x:1219:504:Stoian Genchev,SI,2,5:/home/SI/s61988:/bin/bash
s81430:x:1234:503:Iordan Petkov, KN, k2, g7:/home/KN/s81430:/bin/bash
s61807:x:1248:504:Elica Venchova:/home/SI/s61807:/bin/bash
s62009:x:1254:504:Denitsa Dobрева, 2, 6:/home/SI/s62009:/bin/bash
s61756:x:1258:504:Katrin Kartuleva, SI, 4, 1:/home/SI/s61756:/bin/bash
s855287:x:1195:504:Vaska Kichukova,SI,2,5:/home/SI/s855287:/bin/bash
```

Примерен изход:

```
Katrin Kartuleva:/home/SI/s61756
Elica Venchova:/home/SI/s61807
Stoian Genchev:/home/SI/s61988
Denitsa Dobрева:/home/SI/s62009
Vaska Kichukova:/home/SI/s855287
```

Зад. 11 Вие сте асистент по ОС. На първото упражнение казвате на студентите да си напишат данните на лист, взимате го и им правите акаунти. След упражнението обаче, забравяте да вземете листа със себе си - сещате се половин час по-късно, когато трябва да въведете имената на студентите в

таблица, но за зла беда в стаята вече няма ни помен от листа (вероятно иззет от спешния отряд на GDPR-полицията)

Сещате се, че в началото на упражнението UNIX-часовникът е показвал 1551168000, а в края 1551176100.

Напишете команда, която изкарва разделени с таб факултетните номера и имената на потребителите от специалност СИ, чиито home директории са променили статуса си (status change time) в зададения времеви интервал.

Приемете, че всички потребители от СИ имат home директории под /home/SI.

Примерен изход:

```
62198   Ivaylo Georgiev
62126   Victoria Georgieva
62009   Denitsa Dobрева
62208   Trayana Nedelcheva
```

Няколко реда от /etc/passwd за справка:

```
s62136:x:1302:503:Alexander Ignatov, SI, 2, 2:/home/KN/s62136:/bin/bash
s62171:x:1031:504:Deivid Metanov:/home/SI/s62171:/bin/bash
s62126:x:1016:504:Victoria Georgieva:/home/SI/s62126:/bin/bash
s62009:x:1170:504:Denitsa Dobрева,SI,3,3:/home/SI/s62009:/bin/bash
s62196:x:1221:504:Elena Tuparova,SI,2,1:/home/SI/s62196:/bin/bash
```

Зад. 12 От всички файлове в home директорията на потребителя velin, изведете дълбочината на файл, който:

- има същия inode като този на най-скоро променения файл сред тях
- има минимална дълбочина

Пояснение Под "дълбочина" да се разбира дълбочина в дървото на файловата система: например файлът /foo/bar/baz има дълбочина 3.

Теоретични задачи

Зад. 13 Всеки от процесите P и Q изпълнява поредица от три инструкции:

| process P | process Q |
|-----------|-----------|
| p_1 | q_1 |
| p_2 | q_2 |
| p_3 | q_3 |

Осигурете чрез семафори синхронизация на P и Q така, че инструкцията p_1 да се изпълни преди q_2 , а q_2 да се изпълни преди p_3 .

Зад. 14 Опишете накратко основните процедури и структури данни, необходими за реализация на семафор.

Каква е разликата между слаб и силен семафор?

Опишете максимално несправедлива ситуация, която може да се получи в избирателна секция, ако на входа на секцията пазач – член на изборната комисия пуска гласоподавателите вътре така:

- (1) във всеки момент в секцията може да има най-много двама гласоподаватели.
- (2) пазачът работи като слаб семафор.

Зад. 15

Всеки от процесите P , Q и R изпълнява поредица от три инструкции:

| process P | process Q | process R |
|-----------|-----------|-----------|
|-----------|-----------|-----------|

| | | |
|-----|-----|-----|
| p_1 | q_1 | r_1 |
| p_2 | q_2 | r_2 |
| p_3 | q_3 | r_3 |

Осигурете чрез семафори синхронизация на P , Q и R така, че инструкцията p_1 да се изпълни преди q_2 и r_2 .

Забележка: Решения на задачата с повече от един семафор носят не повече от 20 точки.

Зад. 16 Преди стартиране на процеси P и Q са инициализирани два семафора и брояч:

```
semaphore e, m
e.init(1); m.init(1)
int cnt = 0
```

Паралелно работещи няколко копия на всеки от процесите P и Q изпълняват поредица от инструкции:

| | |
|---------------------|------------|
| process P | process Q |
| m.wait() | e.wait() |
| cnt=cnt+1 | q_section |
| if cnt=1 e.wait() | e.signal() |
| m.signal() | |
| p_section | |
| m.wait() | |
| cnt=cnt-1 | |
| if cnt=0 e.signal() | |
| m.signal() | |

Дайте обоснован отговор на следните въпроси:

- Могат ли едновременно да се изпълняват инструкциите $p_{section}$ и $q_{section}$?
- Могат ли едновременно да се изпълняват няколко инструкции $p_{section}$?
- Могат ли едновременно да се изпълняват няколко инструкции $q_{section}$?
- Има ли условия за deadlock или starvation за някой от процесите?

Упътване:

- Ще казваме, че P е в критична секция, когато изпълнява инструкцията си $p_{section}$. Същото за Q , когато изпълнява $q_{section}$.
- Изяснете смисъла на брояча cnt и какви процеси могат да бъдат приспани в опашките на двата семафора.
- Покажете, че в опашката на семафора e има най-много едно копие на P и произволен брой копия на Q .
- Покажете, че в момента на изпълнение на $e.signal()$ в кой да е от процесите, никой процес не е в критичната си секция.

Зад. 17

- Няколко копия на процеса P изпълняват поредица от три инструкции:

```
process P
p_1
p_2
p_3
```

Осигурете чрез семафор синхронизация на копията така, че най-много един процес да изпълнява инструкция p_2 във всеки един момент.

- Опишете разликата при реализация на слаб и силен семафор.
- Възможно ли е в зависимост от начина на реализация на семафора в подусловие а) да настъпят условия за deadlock или starvation? Ако да, опишете сценарий за поява на неприятната ситуация.

Зад. 18 Всеки от процесите P и Q изпълнява поредица от две инструкции:

| process P | process Q |
|-----------|-----------|
| p_1 | q_1 |
| p_2 | q_2 |

Осигурете чрез семафори синхронизация на P и Q , така че инструкцията p_1 да се изпълни преди q_2 , а q_1 да се изпълни преди p_2 .

Зад. 19 Всеки от процесите P и Q изпълнява поредица от три инструкции:

| process P | process Q |
|-----------|-----------|
| p_1 | q_1 |
| p_2 | q_2 |
| p_3 | q_3 |

Осигурете чрез два семафора синхронизация на P и Q така, че отделните инструкции да се изпълнят в следния времеви ред: $p_1, q_1, p_2, q_2, p_3, q_3$

Зад. 20 Да приемем, че в съвременната операционна система процесът има 4 състояния:

- R – работещ (*running*, използва CPU)
- A – активен (*ready*, очаква CPU)
- S – блокиран (*sleeping*, очаква вход/изход)
- T – изчакващ време (*sleeping*, очаква времеви момент)

Нарисувайте диаграма на състоянията и преходите между тях. Диаграмата е ориентиран граф с върхове отделните състояния и ребра – възможните преходи.

Опишете накратко събитията, предизвикващи преход по всяко ребро на графа.

Зад. 21 Процесът P създава тръба (pipe) с извикване на функцията `pipe(int pipefd[2])` в ОС GNU/Linux.

- а) Кои процеси не могат да ползват тръбата?
- б) Опишете друг метод за изграждане на комуникационен канал, който дава възможност на произволни процеси да изградят и ползват канала. Допълнително искаме новоизградения канал да е достъпен само за процесите, които са го създали.

Упътване: Прочетете man-страницата за функцията `pipe()`.

Зад. 22 Множество паралелно работещи копия на всеки от процесите P и Q изпълняват поредица от две инструкции:

| process P | process Q |
|-----------|-----------|
| p_1 | q_1 |
| p_2 | q_2 |

Осигурете чрез семафори синхронизация на работещите копия, така че:

- а) В произволен момент от времето да работи най-много едно от копията.
- б) Работещите копия да се редуват във времето – след изпълнение на копие на P да следва изпълнение на копие на Q и обратно.
- в) Първоначално е разрешено да се изпълни копие на P .

Зад. 23 Всеки от процесите P , Q и R изпълнява поредица от три инструкции:

| process P | process Q | process R |
|-----------|-----------|-----------|
| p_1 | q_1 | r_1 |
| p_2 | q_2 | r_2 |
| p_3 | q_3 | r_3 |

Осигурете чрез семафори синхронизация на P , Q и R така, че да се изпълнят едновременно следните изисквания:

- Инструкцията p_1 да се изпълни преди q_2 и r_2 .

- Инstrukция r_2 да се изпълни преди p_3 .

Забележка: Решение с 2 семафора ще бъде оценено с 30 точки, решение с повече семафори ще ви донесе 20 точки.

Зад. 24 Всеки от процесите P , Q и R изпълнява поредица от три инструкции:

| process P | process Q | process R |
|-----------|-----------|-----------|
| p_1 | q_1 | r_1 |
| p_2 | q_2 | r_2 |
| p_3 | q_3 | r_3 |

Осигурете чрез семафори синхронизация на P , Q и R така, че да се изпълнят едновременно следните изисквания:

- Инstrukция p_1 да се изпълни преди q_2 .
- Инstrukция q_1 да се изпълни преди r_2 .
- Инstrukция r_1 да се изпълни преди p_2 .
- Инstrukция r_3 да се изпълни след p_2 и q_2 .

Зад. 25 При споделено ползване на памет от няколко процеса е възможно да настъпи надпревара за ресурси (race condition).

(а – 10 точки) Дефинирайте понятието race condition.

(б – 5 точки) Възможно ли е да настъпи race condition в еднопроцесорна система? Ако да, при какви условия.

(в – 15 точки) Какви инструменти ползваме, за да избегнем race condition?

Зад. 26 Една от класическите задачи за синхронизация се нарича *Задача за четателите и писателите* (Readers-writers problem).

- (10 точки) Опишете условието на задачата.
- (20 точки) Опишете решение, използващо семафори.

Зад. 27 Какви са възможните състояния на процес.

Нарисувайте диаграма на състоянията и преходите между тях.

Опишете накратко ситуациите, предизвикващи преходи между състояния.

Зад. 28

Всеки от процесите P , Q и R изпълнява поредица от три инструкции:

| process P | process Q | process R |
|-----------|-----------|-----------|
| p_1 | q_1 | r_1 |
| p_2 | q_2 | r_2 |
| p_3 | q_3 | r_3 |

Осигурете чрез семафори синхронизация на P , Q и R така, че да се изпълнят едновременно следните изисквания:

- Някоя от инструкциите p_2 и q_2 да се изпълни преди r_2 .
- Ако инструкция p_2 се изпълни преди r_2 , то q_2 да се изпълни след r_2 .
- Ако инструкция q_2 се изпълни преди r_2 , то p_2 да се изпълни след r_2 .

Забележка: Решение с 2 семафора ще бъде оценено с 30 точки, решение с повече семафори ще ви донесе 20 точки.

Зад. 29 Дадена е програма за ОС Linux, написана на езика C:

```
#include <unistd.h>
#include <stdio.h>
int main(void)
{
    int p1, p2;
    p1=fork();
```

```

    p2=fork();
    printf("Hello world!\n");
}

```

- а) Колко пъти ще се отпечата текста "Hello world!" при изпълнението на програмата? Обосновете отговора си.
- б) Как работи системното извикване `fork()`?
- в) Нарисувайте кореновото дърво с върхове процесите, които ще се стартират в резултат от изпълнението на програмата и ребра двойките родител-наследник.

Зад. 30 Множество паралелно работещи копия на всеки от процесите P и Q изпълняват поредица от три инструкции:

| process P | process Q |
|-----------|-----------|
| p_1 | q_1 |
| p_2 | q_2 |
| p_3 | q_3 |

Осигурете чрез семафори синхронизация на работещите копия, така че три инструкции – p_1 , q_2 и p_3 се редуват циклично:

- първа се изпълнява инструкция p_1 на някое от работещите копия на процес P ;
- след завършването ѝ се изпълнява инструкция q_2 на някое копие на Q ;
- след нея – p_3 на някое копие на P ;
- с това едно минаване през цикъла завършва и отново може да се изпълни инструкция p_1 на някое от работещите копия на процес P .

Зад. 31 Опишете накратко кои системни извиквания изграждат стандартните комуникационни канали в UNIX – неименувана тръба (pipe), връзка процес-файл, двустранна връзка процес-процес (connection).

Зад. 32 Опишете какви изисквания удовлетворява съвременна файлова система, реализирана върху блоково устройство (*block device*). Опишете накратко реализацията и целта на следните инструменти:

- а) отлагане на запис, алгоритъм на асансьора;
- б) поддържане на журнал на файловата система.

Зад. 33 При реализация на файлова система върху твърд диск файловете и директориите се записват върху сектори от диска. Времето за достъп до секторите зависи от текущото положение на механичните компоненти на диска – над коя пътечка е главата за четене/запис и каква е позицията ѝ над пътечката.

Защо се прави разместване във времето на операциите по четене и запис върху диска?

Опишете накратко реализацията и целта на алгоритъма на асансьора.

Зад. 34 Дадена е програма за ОС Linux, написана на езика C:

```

#include <unistd.h>
#include <stdio.h>
int main(void)
{
    int p1, p2, p3;
    p1=fork();
    if (p1==0) {
        p2=fork();
        if (p2>0) p3=fork();
    }
    printf("Hello world!\n");
}

```

- а) Колко пъти ще се отпечата текста Hello world! при изпълнението на програмата? Обосновете отговора си.
- б) Как работи системното извикване `fork()`?

в) Нарисувайте кореновото дърво с върхове процесите, които ще се стартират в резултат от изпълнението на програмата и ребра двойките родител-наследник.

Зад. 35 Опишете как се изгражда комуникационен канал (connection) между процес-сървер и процес-клиент със следните системни извиквания в стандарта POSIX:

socket(), bind(), connect(), listen(), accept()

Зад. 36 Опишете накратко основните комуникационни канали в ОС Linux.

Кои канали използват пространството на имената и кои не го правят?

Зад. 37 Всеки от процесите P и Q изпълнява поредица от три инструкции:

| process P | process Q |
|-----------|-----------|
| p_1 | q_1 |
| p_2 | q_2 |
| p_3 | q_3 |

Осигурете чрез два семафора синхронизация на P и Q така, че да са изпълнени едновременно следните времеви зависимости:

1. инструкция p_1 да се изпълни преди q_2
2. инструкция q_2 да се изпълни преди p_3
3. инструкция q_1 да се изпълни преди p_2
4. инструкция p_2 да се изпълни преди q_3

Забележка: За решение с повече семафори ще получите 20 точки.

Зад. 38 Множество паралелно работещи копия на всеки от процесите P и Q изпълняват поредица от две инструкции:

| process P | process Q |
|-----------|-----------|
| p_1 | q_1 |
| p_2 | q_2 |

Осигурете чрез семафори синхронизация на работещите копия, така че да са изпълнени едновременно следните условия:

- В произволен момент от времето да работи най-много едно от копията.
- Работещите копия да се редуват във времето – след изпълнение на копие на P да следва изпълнение на копие на Q и обратно.
- Първоначално е разрешено да се изпълни копие на P .

Зад. 39 Процесите P и Q се изпълняват паралелно. Споделената променлива A има начална стойност 4. Променливата R е локална за двата процеса.

| process P | process Q |
|-----------|-----------|
| R=A | R=A |
| R=R+3 | R=R+2 |
| A=R | A=R |

Каква е стойността на A след изпълнението на процесите? Дайте обоснован отговор.

Зад. 40 Опишете разликата между синхронни и асинхронни входно-изходни операции. Дайте примери за програми, при които се налага използването на асинхронен вход-изход.

Зад. 41 Множество паралелно работещи копия на процеса P изпълняват поредица от две инструкции:

process P
p_1
p_2

Осигурете чрез семафори синхронизация на работещите копия, така че:

- Инструкцията p_2 на всяко от работещите копия да се изпълни след като инструкция p_1 е завършила изпълнението си в поне 3 работещи копия.

Упътване: Освен семафори, ползвайте и брояч.

Зад. 42 Опишете реализацията на комуникационна тръба (pipe) чрез семафори. Предполагаме, че тръбата може да съхранява до n байта, подредени в обикновена опашка. Тръбата се ползва от няколко паралелно работещи изпращачи/получатели на байтове. Процесите изпращачи слагат байтове в края на опашката, получателите четат байтове от началото на опашката.