

1 ПОСТАНОВКА ЗАДАЧИ

Основной процесс создает очередь сообщений, после чего ожидает и обрабатывает нажатия клавиш, порождая и завершая процессы двух типов — производители и потребители.

Очередь сообщений представляет собой классическую структуру — кольцевой буфер, содержащий указатели на сообщения, и пара указателей на голову и хвост. Помимо этого очередь содержит счетчик добавленных сообщений и счетчик извлеченных.

Производители формируют сообщения и, если в очереди есть место, перемещают их туда.

Потребители, если в очереди есть сообщения, извлекают их оттуда, обрабатывают и освобождают память с ними связанную.

Для работы используются два семафора для заполнения и извлечения, а также мьютекс или одноместный семафор для монопольного доступа к очереди.

Сообщения имеют следующий формат (размер и смещение в байтах):

Имя	Размер	Смещение	Описание
type	1	0	тип сообщения
hash	2	1	контрольные данные
size	1	3	длина данных в байтах (от 0 до 256)
data	$((\text{size} + 3)/4) * 4$	4	данные сообщения

Производители генерируют сообщения, используя системный генератор `rand(3)` для `size` и `data`. В качестве результата для `size` используется остаток от деления на 257.

Если остаток от деления равен нулю, `rand(3)` вызывается повторно. Если остаток от деления равен 256, значение `size` устанавливается равным 0, реальная длина сообщения при этом составляет 256 байт.

При формировании сообщения контрольные данные формируются из всех байт сообщения. Значение поля `hash` при вычислении контрольных данных принимается равным нулю. Для расчета контрольных данных можно использовать любой подходящий алгоритм на выбор студента.

После помещения значения в очередь перед освобождением мьютекса очереди производитель инкрементирует счетчик добавленных сообщений. Затем после поднятия семафора выводит строку на `stdout`, содержащую помимо всего новое значение этого счетчика.

Потребитель, получив доступ к очереди, извлекает сообщение и удаляет его из очереди. Перед освобождением мьютекса очереди инкрементирует

счетчик извлеченных сообщений. Затем после поднятия семафора проверяет контрольные данные и выводит строку на `stdout`, содержащую помимо всего новое значение счетчика извлеченных сообщений.

При получении сигнала о завершении процесс должен завершить свой цикл и только после этого завершиться, не входя в новый.

Следует предусмотреть задержки, чтобы вывод можно было успеть прочесть в процессе работы программы.

Следует предусмотреть защиту от тупиковых ситуаций из-за отсутствия производителей или потребителей.

2 ТЕСТ

```
spanduy@BILLY-HERRINGTON:~/lab04$ ./main
p
```

[illegible]

```
count - 1
type - 1
hash - 45146
size - 242
data :
138 2      103    158    57     168    172    250    227    27     151    152    44
    109    231    157    26     32     253    42     10     149    240    161    1
    14386  106    227    1      91     109    3      194    12     60     107    184
    54     78     211    205    231    255    59     206    157    85     238    154
    127    24848  112    154    49     255    240    155    227    241    246    80
    244    185    92     48     36     20     103    114    232    52     89     231
    111    39     132197  22     31     68     14     79     180    168    128
    180    152    28     151    137    18     231    125    203    68     174    239
    88     21     98     64     73187  40     185    227    172    126    249    203
    194    7      26     119    176    155    43     72     183    194    210    201
    169    79     149    237    253    13270  18     230    134    92     162    174
    21     133    91     147    126    38     85     133    65     204    53     220
    247    126    147    185    80     92     99159  241    80     157    118    150
    175    92     29     11     254    203    32     131    38     179    1      77
    9      135    142    213    188    106    205    58253  134    138    89     233
    42     75     58     199    193    208    118    29     237    130    28     185
    162    159    223    86     161    44     95     40     18652  228    36     1
    31     33     136    169    123    113    211    198    171    154    135    124
    17     164    105    147    192    34     53     96     2      1391  46     234
    41     233    31     13     13
```

[illegible]

>>>>>>>>>>>>>>>>>>>>PRODUSER>>>>>>>>>>>>>>>>>>>

```
count - 2
type - 0
hash - 6859
size - 45
data :
47      168      214      170      26      169      112      197      68      247      65      85      155
      171      232      92      205      29      188      207      169      189      254      147      230
      231178      243      244      211      32      35      123      246      205      149      159
      61      91      227      52      156      56      208      71
```

[illegible][illegible]

```
count - 3
type - 0
hash - 5391
size - 45
data :
21      62      232      228      231      165      226      122      139      201      45      126      190
      0      158      225      123      148      175      17      52      236      108      23      33
      8      80      241      80      112      29      101      174      5      73      149      170
      44      16      53      245      61      179      179      61
```

[illegible]

C

[illegible]

```
count - 4
type - 0
hash - 17212
size - 150
data :
```


[illegible]

235	168	53	189	50	45	96	79	145	169	233	223	126
	138	12	150	75	196	7	158	115	107	23	11	170
	2148	94	202	57	176	181	225	229	115	19	19	211
	98	164	125	76	132	251	214	144	146	33	84	153
	192	2004	215	211	174	236	3	12	182	60	188	107
	30	162	222	49	181	178	148	89	47	224	221	42
	182	110	188216		194	85	152					

111	93	7	91	97	20	17	157	208	124	187	114	91
	237	39	13	129	129	60	97	94	102	23	204	35
	239	143	120	135	25	210	246	119	217	81	216	237
	98	117	190	223	49	48	58	30	88	71	159	217
	131	0	552	33	23	4	12	7	147	133	142	87
	133	35	48	214	251	30	57	113	220	24	162	12
	82	192	100	153	95	61	28	95	117	5	118	18
	125	12	151									

[illegible]

```
count - 10
type - 0
hash - 24615
size - 185
data :
```

