

# Lab Sheet 1

Name : Lokesh Chandra Basu  
E.No. : 10114026  
Course : Operating Systems Lab

# Question 3

Implement Queue Data Structure. Try to implement various queues like Job Queue, Ready Queue, and Waiting Queue using the data structure developed. Initially, we will use PID(Process ID) (an integer) as the data to be stored in these Queues. You should keep in mind that, in future, they can be modified to store Process Control Block(PCB).

# Algorithm

## Algorithm used :

*//to create the Ready Queue*

*take input from the user for the number of queues in each of the Ready Queue*  
*Initialize count to 0*

*While count is less then input number*

*generate a random number to be stored in queue as pID*

*count = count +1*

*//to create the Job Queue*

*take input from the user for the number of queues in each of the Job Queue*  
*Initialize count to 0*

*While count is less then input number*

*generate a random number to be stored in queue as pID*

*count = count +1*

*//to create the Waiting Queue*

*take input from the user for the number of queues in each of the Waiting Queue*

*Initialize count to 0*

*While count is less then input number*

*generate a random number to be stored in queue as pID*

*count = count +1*

*//to display the Ready Queue*

*initialize count to zero again*

*while count is less then length.readyQueue*

*display info.readyQueue*

*//to display the Job Queue*

*initialize count to zero again*

*while count is less then length.jobQueue*

*display info.jobQueue*

*//to display the Waiting Queue*

*initialize count to zero again*

*while count is less then length.waitQueue*

*display info.waitQueue*

# Gantt Chart

Process	Execution Time(in ms)
P1	5
P2	15
P3	10
P4	7
P5	3

**Job Queue** with processes and execution time (assuming all arrives at same time) :

<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>	<b>P5</b>	
0	5	20	30	37	40

After Long Term Scheduler or Job Scheduler :

**Ready Queue :**

<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>	<b>P5</b>	
0	5	20	30	37	40

*time elapsed = 0ms*

After Short Term Scheduler or Scheduler Dispatch :

**Ready Queue :**

<b>P2</b>	<b>P3</b>	<b>P4</b>	<b>P5</b>	
5	20	30	37	40

**Running Queue :**

<b>P1(Running)</b>	
0	5

*time elapsed = 5ms*

**Ready Queue :**

<b>P3</b>	<b>P4</b>	<b>P5</b>	
20	30	37	40

**Running Queue :**

<b>P1(Terminated)</b>	<b>P2(Running)</b>	
0	5	20

*time elapsed = 20ms (just after P2 terminates)*

*And So on...*

# Output

The screenshot shows a C++ IDE with three source files: `q3.cpp`, `q3.h`, and `q4.cpp`. A terminal window is open, displaying the compilation and execution of `q3.cpp`. The output shows the number of elements in each queue and the process IDs (PIDs) of the processes in each queue.

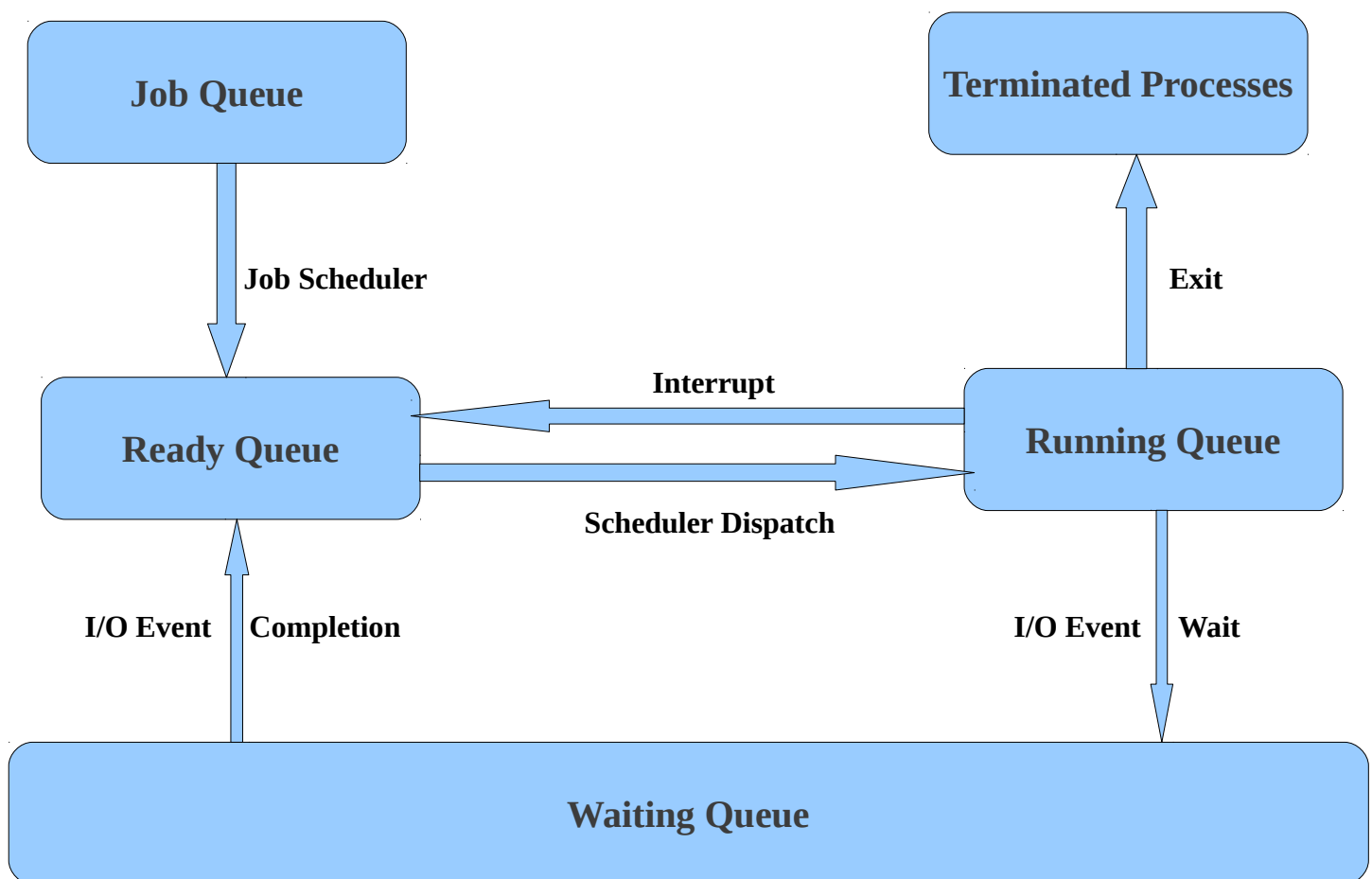
```
lokesh@ubuntu: ~/Dropbox/OS/LS1
lokesh@ubuntu:~/Dropbox/OS/LS1$ g++ q3.cpp -o q3.out
lokesh@ubuntu:~/Dropbox/OS/LS1$ ./q3.out
How many elemets for Ready Queue:
5
How many elemets for Job Queue:
4
How many elemets for Waiting Queue:
6
Ready Queue with process IDs (PIDs): 141 143 195 102 157
Job Queue with process IDs (PIDs): 141 166 124 114
Waiting Queue with process IDs (PIDs): 122 197 150 178 138
lokesh@ubuntu:~/Dropbox/OS/LS1$
```

*gcc version 4.6.3 (Ubuntu/Linaro 4.6.3-1 ubuntu5)*

# Question 4

Simulate (Diagrammatically, if Possible) various Process States. Three key strokes (or buttons) can be used as input to represent a scheduler, an interrupt, and the I/O event. The program should be able use the queues designed in question 3.

## Solution to question 4



# Algorithm

## Algorithm used :

*//to create the Ready Queue*

*take input from the user for the number of queues in each of the Ready Queue*

*Initialize count to 0*

*While count is less then input number*

*generate a random number to be stored in queue as pID*

*count = count +1*

*//to create the Job Queue*

*take input from the user for the number of queues in each of the Job Queue*

*Initialize count to 0*

*While count is less then input number*

*generate a random number to be stored in queue as pID*

*count = count +1*

*//to create the Waiting Queue*

*take input from the user for the number of queues in each of the Waiting Queue*

*Initialize count to 0*

*While count is less then input number*

*generate a random number to be stored in queue as pID*

*count = count +1*

*to display the Menu with all the options:*

**1. Scheduler**

**2. Interrupt**

**3. I/O Event Wait**

**4. I/O Event Completion**

*take input from the user for the operation to perform*

switch choice (1 to 4)

perform the operation according to choice

1. scheduler will dequeue Ready Queue and enqueue Running Queue
2. interrupt will dequeue Running Queue and enqueue Ready Queue
3. I/O Event wait will dequeue Running Queue and enqueue Waiting Queue
4. I/O Event wait will dequeue Waiting Queue and enqueue Ready Queue

if the choice is other then 1 to 4 then the process ends and displays the contents of the queues

*//to display the Ready Queue*

initialize count to zero again

while count is less then length.readyQueue  
display info.readyQueue

*//to display the Job Queue*

initialize count to zero again

while count is less then length.jobQueue  
display info.jobQueue

*//to display the Waiting Queue*

initialize count to zero again

while count is less then length.waitQueue  
display info.waitQueue



# Output

```
Terminal
q3.cpp x q3_output.txt x q3.h x q4.cpp x q4_output.txt x
22 int choice = 0;
23
24
25
26 Queue readyQueue;
27 Queue jobQueue;
28 Queue waitQueue;
29 How many elemets for Ready Queue:
30 3
31 //entry of PIDs
32 cout << "How many elemets for Job Queue:
33 cin >> choice; How many elemets for Waiting Queue:
34 5
35 while (choice != -1)
36 {
37     readyQueue.enqueue((rand()%100)+100);
38     choice--;
39 }
40
41
42 //entry of PIDs
43 cout << "How many elemets for Job Queue:
44 cin >> choice;
45 while (choice != -1)
46 {
47     jobQueue.enqueue((rand()%100)+100);
48     choice--;
49 }
50
51 //entry of PIDs
52 cout << "How many elemets for Waiting Queue:
53 cin >> choice;
54 while (choice != -1)
55 {
56     waitQueue.enqueue((rand()%100)+100);
57     choice--;
58 }
59
60 //entry of PIDs
61 cout << "How many elemets for Ready Queue:
62 cin >> choice;
63 while (choice != -1)
64 {
65     readyQueue.enqueue((rand()%100)+100);
66     choice--;
67 }
68
69 //entry of PIDs
70 cout << "How many elemets for Job Queue:
71 cin >> choice;
72 while (choice != -1)
73 {
74     jobQueue.enqueue((rand()%100)+100);
75     choice--;
76 }
77
78 //entry of PIDs
79 cout << "How many elemets for Waiting Queue:
80 cin >> choice;
81 while (choice != -1)
82 {
83     waitQueue.enqueue((rand()%100)+100);
84     choice--;
85 }
86
87 //entry of PIDs
88 cout << "How many elemets for Ready Queue:
89 cin >> choice;
90 while (choice != -1)
91 {
92     readyQueue.enqueue((rand()%100)+100);
93     choice--;
94 }
95
96 //entry of PIDs
97 cout << "How many elemets for Job Queue:
98 cin >> choice;
99 while (choice != -1)
100 {
101     jobQueue.enqueue((rand()%100)+100);
102     choice--;
103 }
104
105 //entry of PIDs
106 cout << "How many elemets for Waiting Queue:
107 cin >> choice;
108 while (choice != -1)
109 {
110     waitQueue.enqueue((rand()%100)+100);
111     choice--;
112 }
113
114 //entry of PIDs
115 cout << "How many elemets for Ready Queue:
116 cin >> choice;
117 while (choice != -1)
118 {
119     readyQueue.enqueue((rand()%100)+100);
120     choice--;
121 }
122
123 //entry of PIDs
124 cout << "How many elemets for Job Queue:
125 cin >> choice;
126 while (choice != -1)
127 {
128     jobQueue.enqueue((rand()%100)+100);
129     choice--;
130 }
131
132 //entry of PIDs
133 cout << "How many elemets for Waiting Queue:
134 cin >> choice;
135 while (choice != -1)
136 {
137     waitQueue.enqueue((rand()%100)+100);
138     choice--;
139 }
140
141 //entry of PIDs
142 cout << "How many elemets for Ready Queue:
143 cin >> choice;
144 while (choice != -1)
145 {
146     readyQueue.enqueue((rand()%100)+100);
147     choice--;
148 }
149
150 //entry of PIDs
151 cout << "How many elemets for Job Queue:
152 cin >> choice;
153 while (choice != -1)
154 {
155     jobQueue.enqueue((rand()%100)+100);
156     choice--;
157 }
158
159 //entry of PIDs
160 cout << "How many elemets for Waiting Queue:
161 cin >> choice;
162 while (choice != -1)
163 {
164     waitQueue.enqueue((rand()%100)+100);
165     choice--;
166 }
167
168 //entry of PIDs
169 cout << "How many elemets for Ready Queue:
170 cin >> choice;
171 while (choice != -1)
172 {
173     readyQueue.enqueue((rand()%100)+100);
174     choice--;
175 }
176
177 //entry of PIDs
178 cout << "How many elemets for Job Queue:
179 cin >> choice;
180 while (choice != -1)
181 {
182     jobQueue.enqueue((rand()%100)+100);
183     choice--;
184 }
185
186 //entry of PIDs
187 cout << "How many elemets for Waiting Queue:
188 cin >> choice;
189 while (choice != -1)
190 {
191     waitQueue.enqueue((rand()%100)+100);
192     choice--;
193 }
194
195 //entry of PIDs
196 cout << "How many elemets for Ready Queue:
197 cin >> choice;
198 while (choice != -1)
199 {
200     readyQueue.enqueue((rand()%100)+100);
201     choice--;
202 }
203
204 //entry of PIDs
205 cout << "How many elemets for Job Queue:
206 cin >> choice;
207 while (choice != -1)
208 {
209     jobQueue.enqueue((rand()%100)+100);
210     choice--;
211 }
212
213 //entry of PIDs
214 cout << "How many elemets for Waiting Queue:
215 cin >> choice;
216 while (choice != -1)
217 {
218     waitQueue.enqueue((rand()%100)+100);
219     choice--;
220 }
221
222 //entry of PIDs
223 cout << "How many elemets for Ready Queue:
224 cin >> choice;
225 while (choice != -1)
226 {
227     readyQueue.enqueue((rand()%100)+100);
228     choice--;
229 }
230
231 //entry of PIDs
232 cout << "How many elemets for Job Queue:
233 cin >> choice;
234 while (choice != -1)
235 {
236     jobQueue.enqueue((rand()%100)+100);
237     choice--;
238 }
239
240 //entry of PIDs
241 cout << "How many elemets for Waiting Queue:
242 cin >> choice;
243 while (choice != -1)
244 {
245     waitQueue.enqueue((rand()%100)+100);
246     choice--;
247 }
248
249 //entry of PIDs
250 cout << "How many elemets for Ready Queue:
251 cin >> choice;
252 while (choice != -1)
253 {
254     readyQueue.enqueue((rand()%100)+100);
255     choice--;
256 }
257
258 //entry of PIDs
259 cout << "How many elemets for Job Queue:
260 cin >> choice;
261 while (choice != -1)
262 {
263     jobQueue.enqueue((rand()%100)+100);
264     choice--;
265 }
266
267 //entry of PIDs
268 cout << "How many elemets for Waiting Queue:
269 cin >> choice;
270 while (choice != -1)
271 {
272     waitQueue.enqueue((rand()%100)+100);
273     choice--;
274 }
275
276 //entry of PIDs
277 cout << "How many elemets for Ready Queue:
278 cin >> choice;
279 while (choice != -1)
280 {
281     readyQueue.enqueue((rand()%100)+100);
282     choice--;
283 }
284
285 //entry of PIDs
286 cout << "How many elemets for Job Queue:
287 cin >> choice;
288 while (choice != -1)
289 {
290     jobQueue.enqueue((rand()%100)+100);
291     choice--;
292 }
293
294 //entry of PIDs
295 cout << "How many elemets for Waiting Queue:
296 cin >> choice;
297 while (choice != -1)
298 {
299     waitQueue.enqueue((rand()%100)+100);
300     choice--;
301 }
302
303 //entry of PIDs
304 cout << "How many elemets for Ready Queue:
305 cin >> choice;
306 while (choice != -1)
307 {
308     readyQueue.enqueue((rand()%100)+100);
309     choice--;
310 }
311
312 //entry of PIDs
313 cout << "How many elemets for Job Queue:
314 cin >> choice;
315 while (choice != -1)
316 {
317     jobQueue.enqueue((rand()%100)+100);
318     choice--;
319 }
320
321 //entry of PIDs
322 cout << "How many elemets for Waiting Queue:
323 cin >> choice;
324 while (choice != -1)
325 {
326     waitQueue.enqueue((rand()%100)+100);
327     choice--;
328 }
329
330 //entry of PIDs
331 cout << "How many elemets for Ready Queue:
332 cin >> choice;
333 while (choice != -1)
334 {
335     readyQueue.enqueue((rand()%100)+100);
336     choice--;
337 }
338
339 //entry of PIDs
340 cout << "How many elemets for Job Queue:
341 cin >> choice;
342 while (choice != -1)
343 {
344     jobQueue.enqueue((rand()%100)+100);
345     choice--;
346 }
347
348 //entry of PIDs
349 cout << "How many elemets for Waiting Queue:
350 cin >> choice;
351 while (choice != -1)
352 {
353     waitQueue.enqueue((rand()%100)+100);
354     choice--;
355 }
356
357 //entry of PIDs
358 cout << "How many elemets for Ready Queue:
359 cin >> choice;
360 while (choice != -1)
361 {
362     readyQueue.enqueue((rand()%100)+100);
363     choice--;
364 }
365
366 //entry of PIDs
367 cout << "How many elemets for Job Queue:
368 cin >> choice;
369 while (choice != -1)
370 {
371     jobQueue.enqueue((rand()%100)+100);
372     choice--;
373 }
374
375 //entry of PIDs
376 cout << "How many elemets for Waiting Queue:
377 cin >> choice;
378 while (choice != -1)
379 {
380     waitQueue.enqueue((rand()%100)+100);
381     choice--;
382 }
383
384 //entry of PIDs
385 cout << "How many elemets for Ready Queue:
386 cin >> choice;
387 while (choice != -1)
388 {
389     readyQueue.enqueue((rand()%100)+100);
390     choice--;
391 }
392
393 //entry of PIDs
394 cout << "How many elemets for Job Queue:
395 cin >> choice;
396 while (choice != -1)
397 {
398     jobQueue.enqueue((rand()%100)+100);
399     choice--;
400 }
401
402 //entry of PIDs
403 cout << "How many elemets for Waiting Queue:
404 cin >> choice;
405 while (choice != -1)
406 {
407     waitQueue.enqueue((rand()%100)+100);
408     choice--;
409 }
410
411 //entry of PIDs
412 cout << "How many elemets for Ready Queue:
413 cin >> choice;
414 while (choice != -1)
415 {
416     readyQueue.enqueue((rand()%100)+100);
417     choice--;
418 }
419
420 //entry of PIDs
421 cout << "How many elemets for Job Queue:
422 cin >> choice;
423 while (choice != -1)
424 {
425     jobQueue.enqueue((rand()%100)+100);
426     choice--;
427 }
428
429 //entry of PIDs
430 cout << "How many elemets for Waiting Queue:
431 cin &
```

```
Terminal
q3.cpp x q3_output.txt x q3.h x q4.cpp x q4_output.txt x
22 int choice = 0;
23
24
25
26 Queue readyQueue;
27 Queue jobQueue;
28 Queue waitQueue;
29
30
31 //entry of PIDs
32 cout << "How many processes you want to create: ";
33 cin >> choice;
34
35 while (choice != -1)
36 {
37     readyQueue.enqueue(rand() % 1000);
38     choice--;
39 }
40
41 //entry of PIDs
42 cout << "How many times you want to interrupt: ";
43 cin >> choice;
44
45 while (choice != -1)
46 {
47     jobQueue.enqueue(rand() % 1000);
48     choice--;
49 }
50
51 //entry of PIDs
52 cout << "How many times you want to wait for I/O: ";
53 cin >> choice;
54
55 while (choice != -1)
56 {
57     waitQueue.enqueue(rand() % 1000);
58     choice--;
59 }
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
```

**gcc version 4.6.3 (Ubuntu/Linaro 4.6.3-1 ubuntu5)**