

用于支付的方法、装置、电子设备及可读介质

技术领域

[01]本说明书一个或多个实施例涉及物联网技术领域，尤其涉及一种支付方法、装置、电子设备及可读介质。

背景技术

[02]当前，在目标用户购买产品后，可以生成订单二维码，目标用户可以扫描该订单二维码对产品的订单进行支付。

[03]如果当目标用户购买产品时，由于种种原因无法进行支付，可以将产品的订单对应的订单二维码发送给其他用户，其他用户通过扫描订单二维码代替目标用户进行支付。

[04]此时，如果其他用户有多个，可能同时对该订单进行重复支付，重复支付的其他用户需要等系统对订单处理完成后执行退款流程。或者，当订单被首个完成支付的用户支付失败时，需要目标用户刷新操作，系统刷新订单二维码后，让其他用户采用新的订单二维码进行代付。

[05]可见，现有的支付方法在用户进行代付时，订单被成功支付前，订单二维码不能被重复使用同时该订单二维码无法避免重复支付。

发明内容

[06]本说明书一个或多个实施例提供了一种支付方法、装置、电子设备及可读介质。当需要其他用户代替目标用户对订单进行支付时，提供一种在订单被成功支付前订单二维码能够重复使用的同时避免重复支付的支付方法，以减轻系统压力，提高用户体验。

[07]为实现上述目的，本说明书一个或多个实施例提供了一种支付方法，包括：接收第一用户通过扫描订单二维码发送的支付指令；所述订单二维码为根据目标产品对应的第一订单生成的；所述第一订单为基于第二用户对于所述目标产品的购买操作生成的；将所述第一订单设置为占用状态；其中，所述第一订单处于占用状态时拒绝执行所述第一用户之外的用户的支付指令；根据处于所述占用状态的第一订单的订单标识创建所述第一用户的第二订单；基于所述第二订单执行支付操作。

[08]本说明书一个或多个实施例还提供了一种支付装置，包括：接收模块，用于接收第一用户通过扫描订单二维码发送的支付指令；所述订单二维码为根据目标产品对应的第一订单生成的二维码；所述第一订单为基于第二用户对于所述目标产品的购买操作生成的；设置模块，用于将所述第一订单设置为占用状态；其中，所述第一订单处于占用状态时拒绝执行所述第一用户之外的用户的支付指令；创建模块，用于根据处于所述占用状态的第一订单的订单标识创建所述第一用户的第二订单；代付模块，用于基于所述第二订单执行支付操作。

[09]本说明书一个或多个实施例提供的支付方法、装置、电子设备及可读介质，至少具有以下有益效果：本说明书实施例中，首先接收第一用户通过扫描订单二维码发送的支

付指令；所述订单二维码为根据目标产品对应的第一订单生成的；所述第一订单为基于第二用户对于所述目标产品的购买操作生成的；然后将所述第一订单设置为占用状态；其中，所述第一订单处于占用状态时拒绝执行所述第一用户之外的用户的支付指令；之后根据处于所述占用状态的第一订单的订单标识创建所述第一用户的第二订单；最后，基于所述第二订单执行支付操作。可见，由于本说明书实施例接收了支付指令后，会将订单锁定，使得第一用户对第一订单进行支付时，其他用户无法进行支付，且由于是基于第一订单创建的第二订单的进行的支付操作，不直接对第一订单进行处理，即便第一用户支付失败，也仅仅是第二订单支付失败，不影响第一订单，从而订单二维码还可以重复使用，实现了在保证目标产品的第一订单被成功支付前订单二维码能够重复使用的同时避免重复支付的技术效果。

附图说明

[10]为了更清楚地说明本说明书实施例或现有技术中的技术方案，下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅是本说明书中记载的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动性的前提下，还可以根据这些附图获得其他的附图。

[11]图1为本说明书实施例示出的一种支付方法的系统结构示意图。

[12]图2是本说明书实施例示出的一种支付方法的流程示意图。

[13]图3是本说明书实施例示出的一种支付方法的另一种流程示意图。

[14]图4是本说明书实施例示出的一种支付方法的又一种流程示意图。

[15]图5是本说明书实施例示出的一种支付方法的又一种流程示意图。

[16]图6是本说明书实施例示出的一种支付方法的又一种流程示意图。

[17]图7为本说明书实施例示出的一种支付方法的泳道图。

[18]图8为本说明书实施例示出的一种支付装置的结构示意图。

[19]图9为本说明书实施例示出的一种电子设备的结构示意图。

具体实施方式

[20]为使本说明书一个或多个实施例的目的、技术方案和优点更加清楚，下面将结合本说明书具体实施例及相应的附图对本说明书一个或多个实施例的技术方案进行清楚、完整地描述。显然，所描述的实施例仅是本说明书的一部分实施例，而不是全部的实施例。基于本说明中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都属于本说明书一个或多个实施例保护的范围。

[21]以下结合附图，详细说明本说明书各实施例提供的技术方案。

[22]现有技术中，购买者购买产品，例如网购时、或者在线下店面消费时，支付系统或码平台等相关系统确定购买者购买产品后，可以根据该产品的订单生成订单二维码。订单二维码是二维码的一种形式，可以包括金额和产品信息，扫描订单二维码进行支付操作时无需用户键入金额就可以对订单进行支付。当购买者需要代付时，可以将订单二维

码转发给其他用户，由其他用户扫描订单二维码进行代付。

[23]例如将订单二维码发送到群聊中，使得群聊中的其他用户可以代付。此时，如果群聊中用户 A 支付时间过长，同时用户 B 也进行支付，可能出现用户 B 支付成功后，用户 A 也支付成功的情况，即对该产品的订单进行了重复支付。

[24]当前，如果出现了重复支付的情况，需要支付系统进行统一结算时，对用户 A 或用户 B 发起退款操作。不但需要支付系统执行额外的流程，还需要确定退款的目标用户是哪一个，浪费了系统资源。且用户 A 和用户 B 由于均支付成功，可能同时发起退款申请，这还需要由人工协调才能完成准确的退款。显然，重复支付会给系统带来额外的开销且会增加人工的成本，用户体验也很差。

[25]并且，如果用户 A 早于其他用户的支付操作但由于余额不足或其他因素支付失败，由于交易的唯一性，此时该产品的订单对应的订单二维码会失效，其他用户想要代付时，则无法继续采用该订单二维码完成代付，需要购买者主动刷新订单二维码后重新发送给代付者。重新生成订单二维码会给例如支付系统、码平台等相关系统带来额外的开销，用于支付产品订单的订单二维码无法重复使用。

[26]有鉴于此，应当提供一种支付方法，可以重复使用订单二维码的同时，还可以避免重复支付情况的发生，来降低系统的开销，减少用户的额外操作，提高用户体验。

[27]图 1 为本说明书实施例示出的一种支付方法的系统结构示意图。

[28]其中，用户 20 可以通过扫描订单二维码 30 与码平台 10 建立通信连接。当然，用户 20 可以为多个，码平台 10 还可以与统一支付系统相连接。统一支付系统可以为能够使用多种支付平台进行支付的系统。

[29]图 2 是本说明书实施例示出的一种支付方法的流程示意图。

[30]如图 2 所示，本说明书实施例提供的一种支付方法可以包括以下步骤。

[31]步骤 200：接收第一用户通过扫描订单二维码发送的支付指令；所述订单二维码为根据目标产品对应的第一订单生成的；所述第一订单为基于第二用户对于所述目标产品的购买操作生成的；本说明书实施例中，订单二维码可以是与第一订单相对应，当获得第二用户购买目标产品的购买操作时，会触发根据目标产品生成第一订单以及根据第一订单生成与第一订单对应的订单二维码的过程。其中，第二用户可以是在线上例如通过浏览器登录购物网站来购买目标产品，当然，也可以是在线下例如通过自助购买方式购买目标产品。可以理解的是，目标产品可以是实体产品例如电子设备、家居、家电、日常用品、燃油等，也可以是虚拟产品例如游戏点卡、道具、装备、增值服务等，在此不进行限定，只需要能够基于目标产品生成第一订单后，根据第一订单生成订单二维码即可。

[32]在一些实施例中，订单二维码是码平台生成的，码平台可以作为支付系统的一部分，也可以是单独的平台系统。

[33]订单二维码可以包括有目标产品的信息、目标产品的金额、目标产品对应订单的订单号等，具体承载的信息可以根据实际需要进行设置。

[34]第一用户可以采用不同支付平台的应用扫描订单二维码，来为第二用户实现代付。当然，扫描订单二维码的应用不同，应用对应的支付平台也可以不同。

[35]步骤 202：将所述第一订单设置为占用状态；其中，所述第一订单处于占用状态时拒绝执行所述第一用户之外的用户的支付指令；本说明书实施例中，接收到支付指令后，可以将第一订单设置为占用状态。其中，处于占用状态的第一订单只能被占用者进行支付操作，所述第一订单处于占用状态时拒绝执行所述第一用户之外的用户的支付指令。例如，第一用户将第一订单锁定后，第一订单在锁定过程中，只能被第一用户进行支付操作，此时，如果有其他用户例如第三用户试图对第一订单进行支付时，由于第一订单处于占用状态，则不执行第三用户的支付操作。

[36]本说明书实施例中，订单设置为占用状态可以通过数据库的数据行锁实现，也可以是设置参数来实现，例如设置不可占用时间点。

[37]本说明书实施例中，所述将所述第一订单设置为占用状态包括：将所述第一订单所在数据库的数据行锁定。

[38]所述将所述第一订单设置为占用状态包括：将不可占用时间点设置为当前时间点与预设不可占用时间参数的和。

[39]本说明书实施例中，将第一订单设置为占用状态可以包括至少两种不同的方式。

[40]在一些实施例中，可以将第一订单所在数据库的数据行锁定，例如采用数据库行锁将数据行锁定。其中，数据库行锁在同一时间仅允许一个用户进行操作，当第一用户获取数据库行锁对数据行进行锁定时，其他用户无法获取到该数据库行锁，无法执行对数据行锁定的操作，即无法实现对第一订单设置为占用状态的操作。此种方式当并发数较多时，相隔时间非常短的情况下可以实现排他性。

[41]例如，在本申请实施例中的代付场景下，第一用户和第三用户相隔 50ms 分别去获取数据库行锁，以实现对第一订单在数据库中数据行的锁定，完成将第一订单设置为占用状态的操作，由于第一用户发送支付指令的时间点相对于第三用户发送支付指令的时间点靠前，因此，当第三用户试图获取数据库行锁时，可以让第三用户进行等待，也可以直接返回错误信息阻止第三用户的代付操作。

[42]此种方式需要占用数据库，但由于数据库的响应时间短，只需要数据库行锁未被使用，就可以进行锁定，而无需进行其他设置和算法，不用担心网络波动的问题。

[43]另一种实现方式是不占用数据库，而是设置不可占用时间点。所述不可占用时间点用于表征能够执行所述第一用户之外的用户的支付指令的时间点。其中，不可占用时间点表明当前可能有其他用户设置了占用时间，正在进行支付操作。例如，不可占用时间点默认为空或零值，当接收到第一用户发送的支付指令的时间点不早于不可占用时间点时，可以将其设置为当前时间点与预设不可占用时间参数的和。例如，当前时间点为 12 点 0 分 0 秒，预设不可占用时间参数为 2 秒，则将不可占用时间点设置为 12 点 0 分 2 秒。其中，不可占用时间参数可以根据实际需要进行设置。该预设不可占用时间参数可以用于表征处理支付指令的时长。当然，该预设不可占用时间参数可以参考数据库行锁

的使用时长进行设置，可以根据实际时长进行设置，也可以根据后续需要进行更新。

[44]设置不可占用时间点的方式无需占用数据库，可以节省数据库的资源，但是由于设置参数需要执行算法操作，可能由于网络波动而导致设置失败，但可以随着硬件性能和网络性能的不断提升，加快设置速度，避免设置失败。

[45]本说明书实施例中，不管通过哪种方式，均可以将订单设置为占用状态，使得其他用户无法执行支付操作，为避免重复支付进行数据准备。

[46]步骤 204：根据处于所述占用状态的第一订单的订单标识创建所述第一用户的第二订单；本说明书实施例中，能够根据设置为占用状态的第一订单的订单标识创建第一用户的第二订单。其中，订单标识可以根据支付指令中的订单码相匹配得到，该订单码可以使得第一用户、第一订单以及第二订单具有关联关系。

[47]本说明书实施例中，用户不同，对应的第二订单可以不同，第一订单可以对应有多个第二订单，即便第二订单的支付失败或出现错误时，也不影响第一订单的状态，从而不影响第一订单对应的订单二维码的状态，使得订单二维码可以在完成对第一订单的支付前，能够重复使用，无需主动或被动刷新第一订单对应的订单二维码。

[48]当创建了第二订单后，可以在数据库中对创建第二订单的数据表，第一订单的数据表可以对应多个第二订单的数据表，第一订单的数据表与第二订单的数据表之间具有关联关系，例如，具有相同的订单码。每一第二订单对应的用户标识可以不同，来区分不同用户。当然，第二订单的数据表中还可以具有其他与第一订单、用户相关的信息，在此不进行具体限定。

[49]步骤 206：基于所述第二订单执行支付操作。

[50]本说明书实施例中，当生成了第二订单，可以基于第二订单执行支付操作，以完成第一用户对第二用户的代付操作。

[51]本说明书实施例中，能够基于处于占用状态的第一订单标识所创建的第二订单完成支付操作，在第一订单处于占用状态下，即便第二订单支付失败，第一订单对应的订单二维码也可以重复使用，且由于第一订单占用状态下仅能够被一个用户成功支付，在保证订单二维码可以重复使用的情况下可以避免重复支付。

[52]前述实施例中，介绍了将第一订单设置为占用状态的过程，本说明书实施例中，将第一订单设置为占用状态前，还可以先判断第一订单是否处于占用状态。

[53]图 3 是本说明书实施例示出的一种支付方法的另一种流程示意图。

[54]前述实施例的基础上，步骤 202 所述将所述第一订单设置为占用状态前还包括：步骤 300：接收第一用户通过扫描订单二维码发送的支付指令。

[55]步骤 302：判断所述第一订单是否处于占用状态；若确定所述第一订单未处于占用状态，则执行步骤 304。

[56]步骤 304：将所述第一订单设置为占用状态。

[57]步骤 306：根据处于所述占用状态的第一订单的订单标识创建所述第一用户的第二订单。

[58]步骤 308：基于所述第二订单执行支付操作。

[59]步骤 300 以及步骤 304、步骤 306 以及步骤 308 可以参考前述实施例中介绍的步骤 200、步骤 202、步骤 204 以及步骤 206，在此不进行赘述。

[60]其中，所述判断所述第一订单是否处于占用状态具体可以包括：判断所述第一订单所在数据库的数据行是否被锁定；若确定所述数据行未被锁定，确定所述第一订单未处于占用状态。

[61]其中，所述判断所述第一订单是否处于占用状态具体可以包括：判断当前时间点是否早于不可占用时间点；所述不可占用时间点用于表征前一次接收支付指令时所述设置的时间点；若确定所述当前时间点不早于所述不可占用时间点，确定所述第一订单未处于占用状态。

[62]本说明书实施例中，判断第一订单是否处于占用状态，可以是判断第一订单所在数据库的数据行是否被锁定，还可以是判断当前是简单是否早于不可占用时间点。

[63]本说明书实施例中，不可占用时间点是前一次接收支付指令时所设置的时间点。根据前述实施例可以得知，不可占用时间点可以是根据前一次接收到支付指令时，根据该支付指令接收的时间点进行设置的，来表征有用户占用了第一订单，正在处于为第一订单进行支付操作的过程中。

[64]本说明书实施例中，数据行如果被锁定，即数据库行锁有用户已经获取，则意味着有用户正在为第一订单进行支付操作，因此，应当确定第一订单处于占用状态；否则，如果数据库的数据行未被锁定，则确定第一订单未处于占用状态，可以进行后续操作。

[65]本说明书实施例中，可以在将第一订单设置为占用状态前，判断第一订单是否已经被设置为占用状态，若没有设置成占用状态，则将第一订单设置为占用状态。

[66]本说明书实施例中，若确定所述第一订单处于占用状态，则发送表征所述第一订单正在处理中的提示信息给所述第一用户。

[67]本说明书实施例中，如果第一订单处于占用状态，则可以发送表征第一订单正在处理中的提示信息给第一用户，例如，发送“正在处理中”的提示信息，同时，阻止为第一订单进行支付的后续操作流程。

[68]本说明书实施例中，所述确定所述第一订单处于占用状态具体可以包括：若确定所述数据行被锁定，则判断预设等待时间间隔后所述第一订单所在数据库的数据行是否被锁定；若预设等待时间间隔后所述第一订单所在数据库的数据行被锁定，则确定所述第一订单处于占用状态。

[69]本说明书实施例中，如果第一订单处于占用状态，则还可以先等待预设间隔时间。其中预设间隔时间可以根据实际需要进行设置，例如 2 秒。该预设间隔时间可以通过计算从接到支付指令后，生成了第二订单对应的支付码后发送到统一支付平台，接收到统一支付平台返回的例如受理消息等所占用的时间段。可以根据实际需要进行调整。

[70]如果等待预设时间间隔后仍被锁定，则意味着可能存在例如系统问题或网络波动使得前一用户的支付处理未完成，确定第一订单仍处于占用状态。

[71]本说明书实施例中，为了进一步避免重复支付，还能够判断当前时间点是否早于不可支付时间点。

[72]图 4 是本说明书实施例示出的一种支付方法的又一种流程示意图。

[73]步骤 400：接收第一用户通过扫描订单二维码发送的支付指令。

[74]步骤 402：将所述第一订单设置为占用状态。

[75]步骤 404：判断当前时间点是否早于不可支付时间点，若不早于所述不可支付时间点则能够执行步骤 406。

[76]步骤 406：根据处于所述占用状态的第一订单的订单标识创建所述第一用户的第二订单。

[77]步骤 408：基于所述第二订单执行支付操作。

[78]本说明书实施例中，步骤 400、步骤 402 以及步骤 406、步骤 408 可以分别参照前述实施例中的步骤 200、步骤 202、步骤 204 以及步骤 206，在此不进行赘述。

[79]本说明书实施例中，为了进一步避免重复支付，根据处于所述占用状态的第一订单的订单标识创建所述第一用户的第二订单之前，还可以判断当前时间点是否处于其他用户支付操作的时间段内。

[80]本说明书实施例中，步骤 404 可以是当数据库行锁进行解锁后所执行的判断过程，因此，步骤 404 中，判断当前时间点是否早于不可支付时间点之前，还可以判断当前时间点的所述第一订单对应的数据库的数据行是否已经被解锁，如果被解锁则执行步骤 404，否则，等待解锁后执行步骤 404，或者返回错误提示信息给第一用户，例如订单正在被处理中等提示信息。

[81]本说明书实施例中，还需要考虑支付过程中的极端情况。例如，第一用户锁定第一订单的数据库的数据行后，如果第一订单是被其他用户正在支付的过程中，例如第三用户拉起收银台进行支付操作时，此时，虽然第一订单的数据库的数据行已经解除锁定，即数据库行锁已经释放，但是不应当执行后续创建第二订单的步骤，仅应当是在当前时间点不早于不可支付时间点时才能够创建第二订单。

[82]可以理解的是，用户不能长时间处于支付状态中，因此，不可支付时间点也能够保证用户长时间挂起支付操作时而导致其他用户无法进行支付的情况发生。

[83]基于此，本说明书实施例中，还可以判断当前时间点是否早于不可支付时间点。本说明书实施例中，不可支付时间点可以为接收到支付指令时的时间点后的预设支付时间间隔的时间点。还可以是从接收到例如统一支付平台发送的受理消息的时间点后的预设支付时间间隔的时间点。当然，还可以是触发步骤 404 的时间点后的预设支付时间间隔的时间点。

[84]其中，不可支付时间点可以根据实际需要进行设置，例如，用户通常情况从跳转支付页面到支付完成通常不超过 3 分钟的支付操作时长，因此，可以将不可支付时间点设置为从接收支付指令时的时间点后的第 3 分钟作为不可支付时间点，以能够使得用户成功执行支付操作。例如，12 点 00 分后，预设支付时间间隔为 3 分钟，则不可支付时间

点为 12 点 03 分。

[85]当然，如果后续需要改变不可支付时间点，还可以将预设支付时间间隔进行修改。例如修改为 1 分钟，则前述不可支付时间点可以为 12 点 01 分。即，本说明书实施例中的不可支付时间点可以根据实际需要进行设置和更新，是可以反应出用户支付操作时长的参数。

[86]本说明书实施例中，当确定当前时间点不早于不可支付时间点时，确定第一订单为处于支付过程中。可以执行步骤 406 的过程。

[87]可以理解的是，步骤 404 还可以是在步骤 402 之前所执行的过程，根据实际需要步骤 404 的执行顺序可以进行调整。

[88]在上述实施例的基础上，如果确定第一订单处于支付过程中，则可以返回提示信息给第一用户。

[89]本说明书实施例中，还包括：若确定所述当前时间点早于所述不可支付时间点，确定所述第一订单处于支付过程中；将表征第一订单正在被支付的提示信息发送到所述第一用户。

[90]本说明书实施例中，如果确定第一订单处于支付过程中，即第一订单对应的第二订单正在统一支付平台处理的过程中，则可以将提示信息，例如订单正在被支付的提示信息发送到第一用户，并中断后续流程，例如不执行生成第一用户的第二订单的过程。

[91]本说明书实施例中，为了进一步避免重复支付，还可以判断第一订单是否处于支付过程中，并给与用户提示。

[92]本说明书实施例中，还可以判断第一订单是否处于可支付状态，例如是否已经被支付。

[93]图 5 是本说明书实施例示出的一种支付方法的又一种流程示意图。

[94]步骤 500：接收第一用户通过扫描订单二维码发送的支付指令。

[95]步骤 502：判断第一订单是否处于可支付状态，若处于可支付状态则能够执行步骤 504。

[96]步骤 504：将所述第一订单设置为占用状态。

[97]步骤 506：根据处于所述占用状态的第一订单的订单标识创建所述第一用户的第二订单。

[98]步骤 508：基于所述第二订单执行支付操作。

[99]其中，还包括：若确定所述第一订单未处于可支付状态，则将表征所述第一订单已被支付的提示消息发送到所述第一用户。

[100]本说明书实施例中，步骤 500、步骤 504、步骤 506、步骤 508 可以参照前述实施例中的步骤 200、步骤 202、步骤 204 以及步骤 206，在此不进行赘述。

[101]本说明书实施例中，还可以判断第一订单是否处于可支付状态。其中，可支付状态可以表征第一订单能够被执行支付操作。若确定所述第一订单已经被支付，确定所述第一订单处于可支付状态；若确定所述第一订单未被支付，确定所述第一订单未处于可支

付状态；其中，所述第一订单处于可支付状态时，能够根据处于所述占用状态的第一订单的订单标识创建所述第一用户的第二订单。

[102]若确定所述第一订单未处于可支付状态，则将表征所述第一订单已被支付的提示信息发送到所述第一用户。

[103]当第一订单已被支付时，可以将已被支付的提示信息发送给用户。其中，确定第一订单是否已经被支付可以通过第一订单所在数据行中表征是否被支付的字段进行判断，可以通过判断例如是否已被支付的字段来确定第一订单是否已经被支付。还可以是通过第一订单的订单状态进行判断，订单状态可以是订单本申请对应的属性参数，可以通过该属性参数来确定第一订单是否已经被成功支付。

[104]本说明书实施例中，如果第一订单已经被支付，则确定第一订单为处于可支付状态，未处于可支付状态的第一订单不能执行步骤 504-步骤 508 的后续支付流程。处于可支付状态的第一订单能够执行步骤 504-步骤 508 的后续支付流程。

[105]本说明书实施例中，为了节省系统资源，还能够判断第一订单是否处于可支付状态。

[106]当然，判断第一订单是否处于可支付状态还可以通过如下方式实现。

[107]所述根据处于所述占用状态的第一订单的订单标识创建所述第一用户的第二订单之前还包括：若确定所述第一订单对应的交易处于未关闭状态，确定所述第一订单处于可支付状态；若确定所述第一订单对应的交易处于已关闭状态，确定所述第一订单未处于可支付状态。

[108]其中，还包括：若确定所述第一订单未处于可支付状态，则将表征所述第一订单无法被支付的提示信息发送到所述第一用户。

[109]本说明书实施例中，如果第一订单对应的交易处于已关闭状态，则意味着第一订单的交易失效，例如，某秒杀活动已经过期，则第一订单对应的交易处于关闭状态。或者，第二用户主动取消对第一订单的交易，则第一订单对应的交易也应当处于已关闭状态。还可以是其他导致交易状态变动的情况，在此不进行枚举。

[110]本说明书实施例中，如果确定第一订单的交易处于已关闭状态，则可以发送无法被支付的提示信息给第一用户。例如，订单无法被支付，或者订单交易已关闭等。

[111]本说明书实施例中，步骤 502 的执行顺序可以根据实际需要进行设置，以减小系统不必要的开销。

[112]前述实施例中介绍了基于第二订单进行支付操作的过程。下面对此进行详细介绍。

[113]图 6 是本说明书实施例示出的一种支付方法的又一种流程示意图。

[114]步骤 600：接收第一用户通过扫描订单二维码发送的支付指令。

[115]步骤 602：将所述第一订单设置为占用状态。

[116]步骤 604：根据处于所述占用状态的第一订单的订单标识创建所述第一用户的第二订单。

[117]步骤 606：将所述第二订单的支付码发送给统一支付平台；所述统一支付平台能够根据所述第二订单的支付码和所述第一用户扫描所述订单二维码所使用的应用的支

付平台信息，生成支付跳转页面地址信息并发送给所述应用，使得所述第一用户在所述应用根据所述支付跳转页面地址信息弹出的支付界面中，执行支付操作。

[118]本说明书实施例中，步骤 600、步骤 602、步骤 604 的执行过程可以参照前述实施例中步骤 200、步骤 202、步骤 204 的执行过程，在此不进行赘述。

[119]本说明书实施例中，可以支持多种支付平台的支付操作。

[120]为了实现多支付平台的支付操作，本说明书实施例中，可以将第二订单的支付码发送给统一支付平台。其中，支付码可以是依据第二订单生成的，可以包括目标产品的金额、第一用户的标识等相关支付信息。

[121]当将支付码发送给统一支付平台后，统一支付平台可以与第一用户扫描订单二维码所使用的应用建立通信连接，根据应用的支付平台的平台信息，生成对应该支付平台的支付跳转页面，使得第一用户使用该应用根据该支付调整页面地址信息进行支付操作，例如拉起支付收银台等，然后弹出的界面中，执行例如支付确认密码输入等操作。

[122]本说明书实施例中，根据不同的支付平台，可以生成不同的支付跳转页面。来实现不同支付平台的支付。

[123]当然，统一支付平台的功能也可以作为本申请实施例中执行主体的一个模块来实现。

[124]本说明书实施例中，还可以包括：接收所述统一支付平台返回的受理消息；将所述第一订单所在数据库的数据行解锁。

[125]本说明书实施例中，对数据行解锁的时机可以设置为当统一平台受理后。由于数据库本身的限制，如果长时间占用数据行锁，则可能造成数据库运行缓慢、操作日志冗余、数据库崩溃等问题，因此，锁定时间不应太长。例如 3 秒，50 毫秒等。而为了能够保证对第一订单的支付操作，可以在统一平台接收支付码后，解锁数据行。

[126]本说明书实施例中，还可以包括：若接收统一支付平台返回的受理消息后，根据预设支付时间参数更新不可支付时间点。

[127]本说明书实施例中，当接收了统一支付平台返回的受理消息，可以更新不可支付时间点，其中预设支付时间参数可以是参考前述实施例中的预设支付时间间隔设置方式。其可以表征用户对订单的支付操作时长。

[128]本说明书实施例中，还可以包括：当确定第一用户支付成功后，将所述第一订单设置为已支付状态。

[129]本说明书实施例中，还可以异步询问统一支付平台支付情况，如果确定第二订单支付成功，则将第二订单的支付状态修改为已支付，同时将第一订单设置为已支付状态。来完成第一订单的支付。当然，还可以是当支付完成后，统一支付平台主动发送支付结果给本申请的执行主体，例如码平台，然后根据接收到的支付结果确定第一用户是否支付成功。

[130]本说明书实施例中，还可以包括：若确定所述第一用户取消支付操作，获取所述第一用户进行取消支付操作时所对应的取消时间点；若确定所述取消时间点早于不可支付时间点，则将所述不可支付时间点设置为所述取消时间点。

[131]本说明书实施例中,如果第一用户取消了支付操作,为了尽快使得其他用户能够对第一订单进行支付操作,则可以根据第一用户进行取消支付操作的取消时间点来更新不可支付时间点。使得在第一用户支付失败后,尽快使得其他用户能够对第一订单进行支付。

[132]图7为本说明书实施例示出的一种支付方法的泳道图。

[133]参见图7,基于本申请实施例中提供的支付方法,在实际使用中可以包括如下过程:本说明书实施例的执行主体例如码平台10点00分00秒接收第一用户扫描目标产品的订单二维码发送的第一订单的支付指令;获取数据库行锁,将订单行锁定;创建第一订单对应的第二支付单;生成第二订单的支付码,发送到统一支付平台;10点00分30秒接收统一支付平台返回受理通知,将不可支付时间点调整为10点02分;并解锁订单行。

[134]统一支付平台根据第一用户的应用返回支付跳转页面地址,第一用户根据跳转后的界面进行支付操作。

[135]当支付成功后,统一支付平台保存支付结果。当确定该支付结果为成功后,修改第二订单的支付状态为已支付,修改第一订单的支付状态为已支付,第一用户完成支付。

[136]如果10点01分接收了第三用户扫描订单二维码发送支付指令;第一订单的数据库行锁无法获取则反馈给第三用户正在支付的提示信息。

[137]10点03分第三用户发送支付指令,确定第一订单的数据行未锁定,进一步判断第一订单处于已被支付状态,返回第三用户该订单已被支付的提示消息。

[138]可见,本说明书实施例中,能够重复使用该订单二维码的同时避免重复支付。

[139]基于同样的思路,本说明书实施例还提供了上述方法对应的装置。图8为本说明书实施例提供的对应于图2的一种支付装置的结构示意图。

[140]如图8所示,该支付装置可以包括:接收模块81,用于接收第一用户通过扫描订单二维码发送的支付指令;所述订单二维码为根据目标产品对应的第一订单生成的二维码;所述第一订单为基于第二用户对于所述目标产品的购买操作生成的;设置模块82,用于将所述第一订单设置为占用状态;其中,所述第一订单处于占用状态时拒绝执行所述第一用户之外的用户的支付指令;创建模块83,用于根据处于所述占用状态的第一订单的订单标识创建所述第一用户的第二订单;代付模块84,用于基于所述第二订单执行支付操作。

[141]图9为本说明书实施例提供的对应于图2的一种电子设备的结构示意图。如图8所示,电子设备900可以包括:至少一个处理器910;以及,与所述至少一个处理器通信连接的存储器930;其中,所述存储器930存储有可被所述至少一个处理器910执行的指令920,所述指令被所述至少一个处理器910执行,以使所述至少一个处理器910能够:接收第一用户通过扫描订单二维码发送的支付指令;所述订单二维码为根据目标产品对应的第一订单生成的;所述第一订单为基于第二用户对于所述目标产品的购买操作生成的;将所述第一订单设置为占用状态;其中,所述第一订单处于占用状态时拒绝执

行所述第一用户之外的用户的支付指令；根据处于所述占用状态的第一订单的订单标识创建所述第一用户的第二订单；基于所述第二订单执行支付操作。

[142]本说明书中的各个实施例均采用递进的方式描述，各个实施例之间相同相似的部分互相参见即可，每个实施例重点说明的都是与其他实施例的不同之处。尤其，对于图7所示的支付装置而言，由于其基本相似于方法实施例，所以描述的比较简单，相关之处参见方法实施例的部分说明即可。

[143]在20世纪90年代，对于一个技术的改进可以很明显地区分是硬件上的改进(例如，对二极管、晶体管、开关等电路结构的改进)还是软件上的改进(对于方法流程的改进)。然而，随着技术的发展，当今的很多方法流程的改进已经可以视为硬件电路结构的直接改进。设计人员几乎都通过将改进的方法流程编程到硬件电路中来得到相应的硬件电路结构。因此，不能说一个方法流程的改进就不能用硬件实体模块来实现。例如，可编程逻辑器件(Programmable Logic Device, PLD)(例如现场可编程门阵列(Field Programmable Gate Array, FPGA))就是这样一种集成电路，其逻辑功能由用户对器件编程来确定。由设计人员自行编程来把一个数字系统“集成”在一片PLD上，而不需要请芯片制造厂商来设计和制作专用的集成电路芯片。而且，如今，取代手工地制作集成电路芯片，这种编程也多半改用“逻辑编译器(logic compiler)”软件来实现，它与程序开发撰写时所用的软件编译器相类似，而要编译之前的原始代码也得用特定的编程语言来撰写，此称之为硬件描述语言(Hardware Description Language, HDL)，而HDL也并非仅有一种，而是有许多种，如ABEL(Advanced Boolean Expression Language)、AHDL(Altera Hardware Description Language)、Confluence、CUPL(Cornell University Programming Language)、HDCal、JHDL(Java Hardware Description Language)、Lava、Lola、MyHDL、PALASM、RHDL(Ruby Hardware Description Language)等，目前最普遍使用的是VHDL(Very-High-Speed Integrated Circuit Hardware Description Language)与Verilog。本领域技术人员也应该清楚，只需要将方法流程用上述几种硬件描述语言稍作逻辑编程并编程到集成电路中，就可以很容易得到实现该逻辑方法流程的硬件电路。

[144]控制器可以按任何适当的方式实现，例如，控制器可以采取例如微处理器或处理器以及存储可由该(微)处理器执行的计算机可读程序代码(例如软件或固件)的计算机可读介质、逻辑门、开关、专用集成电路(Application Specific Integrated Circuit, ASIC)、可编程逻辑控制器和嵌入微控制器的形式，控制器的例子包括但不限于以下微控制器：ARC 625D、Atmel AT91SAM、Microchip PIC18F26K20 以及Silicone Labs C8051F320，存储器控制器还可以被实现为存储器的控制逻辑的一部分。本领域技术人员也知道，除了以纯计算机可读程序代码方式实现控制器以外，完全可以通过将方法步骤进行逻辑编程来使得控制器以逻辑门、开关、专用集成电路、可编程逻辑控制器和嵌入微控制器等的形式来实现相同功能。因此这种控制器可以被认为是一种硬件部件，而对其内包括的用于实现各种功能的装置也可以视为硬件部件内的结构。或者甚至，可以将用于实现各种功能的装置视为既可以是实现方法的软件模块又可以是硬件部件内的结构。

[145]上述实施例阐明的系统、装置、模块或单元，具体可以由计算机芯片或实体实现，或者由具有某种功能的产品来实现。一种典型的实现设备为计算机。具体的，计算机例如可以为个人计算机、膝上型计算机、蜂窝电话、相机电话、智能电话、个人数字助理、媒体播放器、导航设备、电子邮件设备、游戏控制台、平板计算机、可穿戴设备或者这些设备中的任何设备的组合。

[146]为了描述的方便，描述以上装置时以功能分为各种单元分别描述。当然，在实施本申请时可以把各单元的功能在同一个或多个软件和/或硬件中实现。

[147]本领域内的技术人员应明白，本说明书的实施例可提供为方法、系统、或计算机程序产品。因此，本申请可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实施例的形式。而且，本申请可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质（包括但不限于磁盘存储器、CD-ROM、光学存储器等）上实施的计算机程序产品的形式。

[148]本申请是参照根据本说明书实施例的方法、设备（系统）、和计算机程序产品的流程图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器，使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的装置。

[149]这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理设备以特定方式工作的计算机可读存储器中，使得存储在该计算机可读存储器中的指令产生包括指令装置的制造品，该指令装置实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能。

[150]这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上，使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理，从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的步骤。

[151]在一个典型的配置中，计算设备包括一个或多个处理器(CPU)、输入/输出接口、网络接口和内存。

[152]内存可能包括计算机可读介质中的非永久性存储器，随机存取存储器(RAM)和/或非易失性内存等形式，如只读存储器(ROM)或闪存(flash RAM)。内存是计算机可读介质的示例。

[153]计算机可读介质包括永久性和非永久性、可移动和非可移动媒体可以由任何方法或技术来实现信息存储。信息可以是计算机可读指令、数据结构、程序的模块或其他数据。计算机的存储介质的例子包括，但不限于相变内存(PRAM)、静态随机存取存储器(SRAM)、动态随机存取存储器(DRAM)、其他类型的随机存取存储器(RAM)、只读存储

器(ROM)、电可擦除可编程只读存储器(EEPROM)、快闪记忆体或其他内存技术、只读光盘只读存储器(CD-ROM)、数字多功能光盘(DVD)或其他光学存储、磁盒式磁带、磁带式磁盘存储或其他磁性存储设备或任何其他非传输介质,可用于存储可以被计算设备访问的信息。按照本文中的界定,计算机可读介质不包括暂存电脑可读媒体(transitory media),如调制的数据信号和载波。

[154]还需要说明的是,术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含,从而使得包括一系列要素的过程、方法、商品或者设备不仅包括那些要素,而且还包括没有明确列出的其他要素,或者是还包括为这种过程、方法、商品或者设备所固有的要素。在没有更多限制的情况下,由语句“包括一个……”限定的要素,并不排除在包括所述要素的过程、方法、商品或者设备中还存在另外的相同要素。

[155]本领域技术人员应明白,本申请的实施例可提供为方法、系统或计算机程序产品。因此,本申请可采用完全硬件实施例、完全软件实施例或结合软件和硬件方面的实施例的形式。而且,本申请可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质(包括但不限于磁盘存储器、CD-ROM、光学存储器等)上实施的计算机程序产品的形式。

[156]本申请可以在由计算机执行的计算机可执行指令的一般上下文中描述,例如程序模块。一般地,程序模块包括执行特定任务或实现特定抽象数据类型的例程、程序、对象、组件、数据结构等等。也可以在分布式计算环境中实践本申请,在这些分布式计算环境中,由通过通信网络而被连接的远程处理设备来执行任务。在分布式计算环境中,程序模块可以位于包括存储设备在内的本地和远程计算机存储介质中。

[157]以上所述仅为本申请的实施例而已,并不用于限制本申请。对于本领域技术人员来说,本申请可以有各种更改和变化。凡在本申请的精神和原理之内所作的任何修改、等同替换、改进等,均应包含在本申请的权利要求范围之内。

权利要求书

1、一种支付方法，包括：

接收第一用户通过扫描订单二维码发送的支付指令；所述订单二维码为根据目标产品对应的第一订单生成的；所述第一订单为基于第二用户对于所述目标产品的购买操作生成的；

将所述第一订单设置为占用状态；其中，所述第一订单处于占用状态时拒绝执行所述第一用户之外的用户的支付指令；

根据处于所述占用状态的第一订单的订单标识创建所述第一用户的第二订单；

基于所述第二订单执行支付操作。

2、根据权利要求1所述的支付方法，所述将所述第一订单设置为占用状态前还包括：

判断所述第一订单是否处于占用状态；

若确定所述第一订单未处于占用状态，则执行所述将所述第一订单设置为占用状态的步骤。

3、根据权利要求2所述的支付方法，所述判断所述第一订单是否处于占用状态具体包括：

判断所述第一订单所在数据库的数据行是否被锁定；

若确定所述数据行未被锁定，确定所述第一订单未处于占用状态。

4、根据权利要求2所述的支付方法，所述判断所述第一订单是否处于占用状态具体包括：

判断当前时间点是否早于不可占用时间点；所述不可占用时间点用于表征前一次接收支付指令时所述设置的时间点；

若确定所述当前时间点不早于所述不可占用时间点，确定所述第一订单未处于占用状态。

5、根据权利要求1所述的支付方法，所述将所述第一订单设置为占用状态包括：

将所述第一订单所在数据库的数据行锁定。

6、根据权利要求1所述的支付方法，所述将所述第一订单设置为占用状态包括：

将不可占用时间点设置为当前时间点与预设不可占用时间参数的和；所述不可占用时间点用于表征能够执行所述第一用户之外的用户的支付指令的时间点。

7、根据权利要求2所述的支付方法，若确定所述第一订单处于占用状态，则发送表征所述第一订单正在处理中的提示信息给所述第一用户。

8、根据权利要求3所述的支付方法，所述判断所述第一订单是否处于占用状态还包括：

若确定所述数据行被锁定，则判断预设等待时间间隔后所述第一订单所在数据库的数据行是否被锁定；

若预设等待时间间隔后所述第一订单所在数据库的数据行被锁定，则确定所述第一

订单处于占用状态。

9、根据权利要求 1 至 8 任意一项所述的支付方法，所述根据处于所述占用状态的第一订单的订单标识创建所述第一用户的第二订单之前，还包括：

判断所述当前时间点是否早于不可支付时间点；

若确定所述当前时间点不早于所述不可支付时间点，确定所述第一订单未处于支付过程中；

其中，当所述第一订单未处于支付过程中时，能够执行根据处于所述占用状态的第一订单的订单标识创建所述第一用户的第二订单的操作。

10、根据权利要求 9 所述的支付方法，还包括：

若确定所述当前时间点早于所述不可支付时间点，确定所述第一订单处于支付过程中；

将表征第一订单正在被支付的提示信息发送到所述第一用户。

11、根据权利要求 1 所述的支付方法，所述根据处于所述占用状态的第一订单的订单标识创建所述第一用户的第二订单之前还包括：

若确定所述第一订单已经被支付，确定所述第一订单处于可支付状态；

若确定所述第一订单未被支付，确定所述第一订单未处于可支付状态；

其中，所述第一订单处于可支付状态时，能够根据处于所述占用状态的第一订单的订单标识创建所述第一用户的第二订单。

12、根据权利要求 1 所述的支付方法，所述根据处于所述占用状态的第一订单的订单标识创建所述第一用户的第二订单之前还包括：

若确定所述第一订单对应的交易处于未关闭状态，确定所述第一订单处于可支付状态；

若确定所述第一订单对应的交易处于已关闭状态，确定所述第一订单未处于可支付状态。

13、根据权利要求 11 所述的支付方法，还包括：

若确定所述第一订单未处于可支付状态，则将表征所述第一订单已被支付的提示信息发送到所述第一用户。

14、根据权利要求 12 所述的支付方法，还包括：

若确定所述第一订单未处于可支付状态，则将表征所述第一订单无法被支付的提示信息发送到所述第一用户。

15、根据权利要求 1 所述的支付方法，所述基于所述第二订单执行支付操作具体包括：

将所述第二订单的支付码发送给统一支付平台；所述统一支付平台能够根据所述第二订单的支付码和所述第一用户扫描所述订单二维码所使用的应用的支付平台信息生成支付跳转页面地址信息并发送给所述应用，使得所述第一用户在所述应用根据所述支付跳转页面地址信息弹出的支付界面中执行支付操作。

16、根据权利要求 15 所述的支付方法，还包括：

接收所述统一支付平台返回的受理消息；

将所述第一订单所在数据库的数据行解锁。

17、根据权利要求 9 所述的支付方法，还包括：

若接收统一支付平台返回的受理消息后，根据预设支付时间参数更新不可支付时间点。

18、根据权利要求 1 所述的支付方法，还包括：

当确定第一用户支付成功后，将所述第一订单设置为已支付状态。

19、根据权利要求 1 所述的支付方法，还包括：

若确定所述第一用户取消支付操作，获取所述第一用户进行取消支付操作时所对应的取消时间点；

若确定所述取消时间点早于不可支付时间点，则将所述不可支付时间点设置为所述取消时间点。

20、根据权利要求 1 所述的支付方法，所述第一订单的订单标识与所述第二订单的订单标识具有对应关系。

21、一种支付装置，包括：

接收模块，用于接收第一用户通过扫描订单二维码发送的支付指令；所述订单二维码为根据目标产品对应的第一订单生成的二维码；所述第一订单为基于第二用户对于所述目标产品的购买操作生成的；

设置模块，用于将所述第一订单设置为占用状态；其中，所述第一订单处于占用状态时拒绝执行所述第一用户之外的用户的支付指令；

创建模块，用于根据处于所述占用状态的第一订单的订单标识创建所述第一用户的第二订单；

代付模块，用于基于所述第二订单执行支付操作。

22、一种电子设备，包括：

至少一个处理器；以及，

与所述至少一个处理器通信连接的存储器；其中，

所述存储器存储有可被所述至少一个处理器执行的指令，所述指令被所述至少一个处理器执行，以使所述至少一个处理器能够：

接收第一用户通过扫描订单二维码发送的支付指令；所述订单二维码为根据目标产品对应的第一订单生成的二维码；所述第一订单为基于第二用户对于所述目标产品的购买操作生成的；

将所述第一订单设置为占用状态；其中，所述第一订单处于占用状态时拒绝执行所述第一用户之外的用户的支付指令；

根据处于所述占用状态的第一订单的订单标识创建所述第一用户的第二订单；

基于所述第二订单执行支付操作。

23、一种计算机可读介质，其上存储有计算机可读指令，所述计算机可读指令可被处理器执行以实现权利要求 1 至 20 中任一项所述的一种支付方法。