

```

;; 鏈上費用計算

;; 繁體中文註解版 by Y.C.
;; 所有中文均經過人腦編譯，不存在 AI 製造的奇怪語法，並可取代原有的 op-code.fc 及
workchain.fc 服用。


;; 如複製此頁貼上到原有項目，需要修改路徑至工具庫擺放的資料夾位置；
;; 在原版文檔中，op-code_workchain_zh.fc 內容分別以 op-code.fc 及 workchain.fc 擺放。
#include "op-code_workchain_zh.fc";


;; 定義 1 TON 為 1,000,000,000 nano-TON (也就是 1 TON)
const ONE_TON = 1000000000;


;; 把智能合約帳戶的最小存儲時間設定為 5 年（轉換為秒數）
const MIN_STORAGE_DURATION = 5 * 365 * 24 * 3600; ;; 5 years


{-

# 預編譯常量
- 以下內容均為合約模擬測試的結果


## 最小手續費
- 轉帳手續費 [/sandbox_tests/JettonWallet.spec.ts#L935] (L935) `0.028627415`
TON
- 燃燒手續費 [/sandbox_tests/JettonWallet.spec.ts#L1185] (L1185) `0.016492002`
TON


## 存儲費用
- 存儲費用在單獨的測試文件中計算
[/sandbox_tests/StateInit.spec.ts] (StateInit.spec.ts)


在 JETTON_WALLET_BITS 和 JETTON_WALLET_INITSTATE_BITS 之間的區別
正正是 StateInit 和 AccountStorage 的差異
詳情請參考
https://github.com/ton-blockchain/ton/blob/master/crypto/block/block.tlb
這裡按照最大可能的餘額計算位元的數量。
而 Gas 常量通過主測試套件計算；先找到相關交易，再將其 gas 消耗結果顯示到 console。

-}

```

```

;; jetton-wallet.fc#L163 - 手動計算位元數
const BURN_NOTIFICATION_BITS          = 754; ;; body =
32+64+124+(3+8+256)+(3+8+256)

const BURN_NOTIFICATION_CELLS         = 1;    ;; body 始終存放於引用中

;; Jetton 錢包佔用的位元數 [/sandbox_tests/StateInit.spec.ts#L92] (L92)
const JETTON_WALLET_BITS               = 1033;

;; Jetton 錢包佔用的 cell 數 [/sandbox_tests/StateInit.spec.ts#L92] (L92)
const JETTON_WALLET_CELLS              = 3;

;; Jetton 錢包的初始狀態位元數 [/sandbox_tests/StateInit.spec.ts#L95] (L95)
const JETTON_WALLET_INITSTATE_BITS     = 931;

;; Jetton 錢包的初始狀態的 cell 數量 [/sandbox_tests/StateInit.spec.ts#L95] (L95)
const JETTON_WALLET_INITSTATE_CELLS    = 3;

;; 發送轉帳的 Gas 消耗 [/sandbox_tests/JettonWallet.spec.ts#L853] (L853)
const SEND_TRANSFER_GAS_CONSUMPTION    = 9255;

;; 接收轉帳的 Gas 消耗 [/sandbox_tests/JettonWallet.spec.ts#L862] (L862)
const RECEIVE_TRANSFER_GAS_CONSUMPTION = 10355;

;; 發送燃燒的 Gas 消耗 [/sandbox_tests/JettonWallet.spec.ts#L1154] (L1154)
const SEND_BURN_GAS_CONSUMPTION        = 5791;

;; 接收燃燒的 Gas 消耗 [/sandbox_tests/JettonWallet.spec.ts#L1155] (L1155)
const RECEIVE_BURN_GAS_CONSUMPTION     = 6775;

;;
-----

;; 計算 Jetton 錢包的最小存儲費用
int calculate_jetton_wallet_min_storage_fee() inline {
    ;; 引用 stdlib.fc / stdlib_zh.fc 當中的:
    ;; int get_storage_fee(int workchain, int seconds, int bits, int cells)
    asm(cells bits seconds workchain) "GETSTORAGEFEE";
    return get_storage_fee(MY_WORKCHAIN, MIN_STORAGE_DURATION, JETTON_WALLET_BITS,
        JETTON_WALLET_CELLS);
}

```

;; 計算初始狀態轉發的額外費用

```
int forward_init_state_overhead() inline {  
    ;; 引用 stdlib.fc / stdlib_zh.fc 當中的：  
    ;; int get_simple_forward_fee(int workchain, int bits, int cells) asm(cells bits workchain)  
    "GETFORWARDFEESIMPLE";  
    return get_simple_forward_fee(MY_WORKCHAIN, JETTON_WALLET_INITSTATE_BITS,  
    JETTON_WALLET_INITSTATE_CELLS);  
}
```

;; 檢查轉帳金額是否足夠進行轉帳

```
( ) check_amount_is_enough_to_transfer(int msg_value, int forward_ton_amount, int  
fwd_fee) impure inline {  
    ;; 如果有需要轉發的 TON, 則設置轉發次數為 2, 否則為 1。  
    int fwd_count = forward_ton_amount ? 2 : 1;  
    ;; 獲取 Jetton 錢包的 Gas 消耗, 若無預編譯數據則使用默認值。  
    int jetton_wallet_gas_consumption = get_precompiled_gas_consumption();  
    int send_transfer_gas_consumption = null?(jetton_wallet_gas_consumption)  
        ? SEND_TRANSFER_GAS_CONSUMPTION : jetton_wallet_gas_consumption;  
    int receive_transfer_gas_consumption = null?(jetton_wallet_gas_consumption)  
        ? RECEIVE_TRANSFER_GAS_CONSUMPTION : jetton_wallet_gas_consumption;  
    ;; 檢查 msg_value 是否足夠支付所有轉帳相關費用。  
    throw_unless(error::not_enough_gas, msg_value >  
    forward_ton_amount +  
    ;; 轉帳涉及 3 條消息: 從 wal1 -> wal2, wal2 -> owner, wal2 -> response  
    ;; 最後一條消息是可選的, 如果失敗也沒問題。  
    fwd_count * fwd_fee +  
    forward_init_state_overhead() + ;; 與 initstate 相關的額外轉發費用  
    get_compute_fee(MY_WORKCHAIN, send_transfer_gas_consumption) +  
    get_compute_fee(MY_WORKCHAIN, receive_transfer_gas_consumption) +  
    calculate_jetton_wallet_min_storage_fee() );  
}
```

;; 檢查燃燒操作的金額是否足夠

```
( ) check_amount_is_enough_to_burn(int msg_value) impure inline {  
    int jetton_wallet_gas_consumption = get_precompiled_gas_consumption();  
    int send_burn_gas_consumption = null?(jetton_wallet_gas_consumption)  
        ? SEND_BURN_GAS_CONSUMPTION : jetton_wallet_gas_consumption;  
    ;; 檢查 msg_value 是否足夠支付燃燒操作的轉發費用和 gas 費用  
    throw_unless(error::not_enough_gas, msg_value > get_forward_fee(MY_WORKCHAIN,  
    BURN_NOTIFICATION_BITS,  
    BURN_NOTIFICATION_CELLS) + get_compute_fee(MY_WORKCHAIN, send_burn_gas_consumption)  
    + get_compute_fee(MY_WORKCHAIN, RECEIVE_BURN_GAS_CONSUMPTION));  
}
```