

## 観光地の食堂で長く待たされるのはなぜ？

畝 徳治

先日、観光客で賑わう隣町へ、遠くから訪ねてくれた友人達と一緒に紅葉見物に出掛けた。どこかで昼食を摂ろうと適当な店を探していた。どこも長い列が店の外まで続いている。ある店に決めた後、友人の一人が言った。「畝さん、私の分の食券を買ってきてくれない。私が座席を探しておくから。」と。私は答えた。「先に座席を確保すると、反って自分を苦しめることがあるよ。」と。

案の定、その店の状態は次のようになっていた。



- ① 幾つかの席の客は食事をしているが、その他の多くの客席には食べ物がまだ無い。その様子から見れば、仲間が食券を購入するのを客席で待っている状態。食券が購入できていないので、その人たちの分の調理は未だ開始していない。
- ② 厨房では、客席が混雑している割には調理人が手持無沙汰でいる。調理人は新たな注文が来るのを待っているだけで、何の調理もしていない。



- ③ 食券売り場では、来客が食券を求めて長い列を作っている。しかし、客席が満杯なので新たな食券販売を止めている。



この状態では、目標（①客はできるだけ早く食事を済ませること、②調理人はできるだけ早く食事を作ること、③レジ係はできるだけ早くの食券を売ること。）に向かって進行しているのは、始めの頃に食事を注文し食事を開始している少数の客だけである。殆んどの人(多数の客、調理人、レジ係)は、互いに待ち状態に陥っている。

食券を購入しようとして待つ客 A、注文が来るのを待つ調理人 B、客席が空くのを待つレジ係 C の三者だけならば完全なデッドロック（※）になる。一部の客席で食事をしている客がいるので、完全なデッドロックではない。完全なデッドロックなら問題が生じていることに気づくと思われるが、一部の客席では目標に向かって進行（食事を）しているので、何となく「客席の回転が遅い」と感じて、問題が生じていることに気づかないようだ。

この店の調理能力の半分以下しか活かせていない気がした。店が混雑していた割には、この日の売り上げは少なかったに違いない。

対策を考えるために少し詳しく見ましょう。食券を買おうとしている客に着目すると、買えない理由は何でしょうか。客席が満杯になっているからです。満杯にしているのは誰でしょうか。先客、競合する客、自分の仲間と一緒に客席を塞いでいるからです。極端に言えば、自分の仲間が客席を塞いでいるがために自分たちの食券が買えない訳です。

無意味な待ち状態を発生させないためにはどうすれば良いでしょうか。答えは、各客の客席確保と注文（調理開始）を同期させることです。即ち、食券を持たない客が客席に着くのを制止することです。そうすれば、客席に着いている客は、注文済みであり調理が開始されているので、調理に掛かる時間を待てば必ず食事が始められます。食事が済めば客席を解放できます。

ビジネス街の店では、①席に着いた客から注文を取る、または②食券を購入（注文）してから席に着かせるので、このような輻輳はあまり生じません。

観光客は、家族や仲間と共に行動し、互いに協力しあう。一人が代表して食券を購入し、その他の人たちが別動で客席を確保しようとする。空席が見える程度に店が空いている状況ならば、そうしても全く問題がない。しかし、空席が無くなるほどに混雑すると、仲間同士が協力するという美談が、このような無用の輻輳状態を発生させる原因になる。

店の人は食券を持たない客が席に着くのを躊躇せず制止し、客は慌てて席取りをしなことがお互いの利益になります。これが社会の常識になると観光地の食事がもっと早く済ませられると思います。多くの方に知って欲しいことです。

彼の友人は客席の状態を見て言っていました。「確かに畝さんが言った状態になっている。」と。そして、食事を諦めて店を出た後で言っていました。「これまで自分は店が混んでいると、直ぐに手分けをするように提案していた。それが害になるとは知らなかった。」「これからは注意するよ。」と。

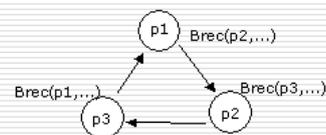
### ※デッドロック 【 deadlock 】

コンピューターで複数のプログラムが動作している際、プログラム間で互いの処理終了を待つなどして、半永久的な待機状態になること。

並行プロセスの挙動

### デッドロックとは？

- システムを構成しているすべてのプロセスが起きるはずのない事象を待ち続ける状態



© H. Tokuda 2016