

コードレビュー

Vol.4

USP 研究所技術研究員
written by 大内智明

CODE Review

前号に続き、本部・店舗間の発注・伝票処理に伴う連絡・通知をシンプルに実現する常駐型シェルスクリプトをご紹介します。この処理フローの一部を担う集積処理に今回はフォーカスします。

定期的にファイルを集積する

本部や店舗で行われる業務には、決まった締め日時があります。その時期になると本部や各店舗で多くのトランザクションが集中的に発生します。結果として、システム負荷が大きくなり、他システムへの連携に影響する場合があります。こうしたシステム負荷が大きくなるように、ユニケーij開発手法でよく使用するのが、前回登場した常駐型シェルスクリプトです。

今回は、常駐型シェルスクリプトの全体像を説明しましたが、今回はその中の重要な一機能である集積処理部分のコードを見ていきます（図1）。集積処理の役割は、各業務サーバーで発生した複数のトランザクシ

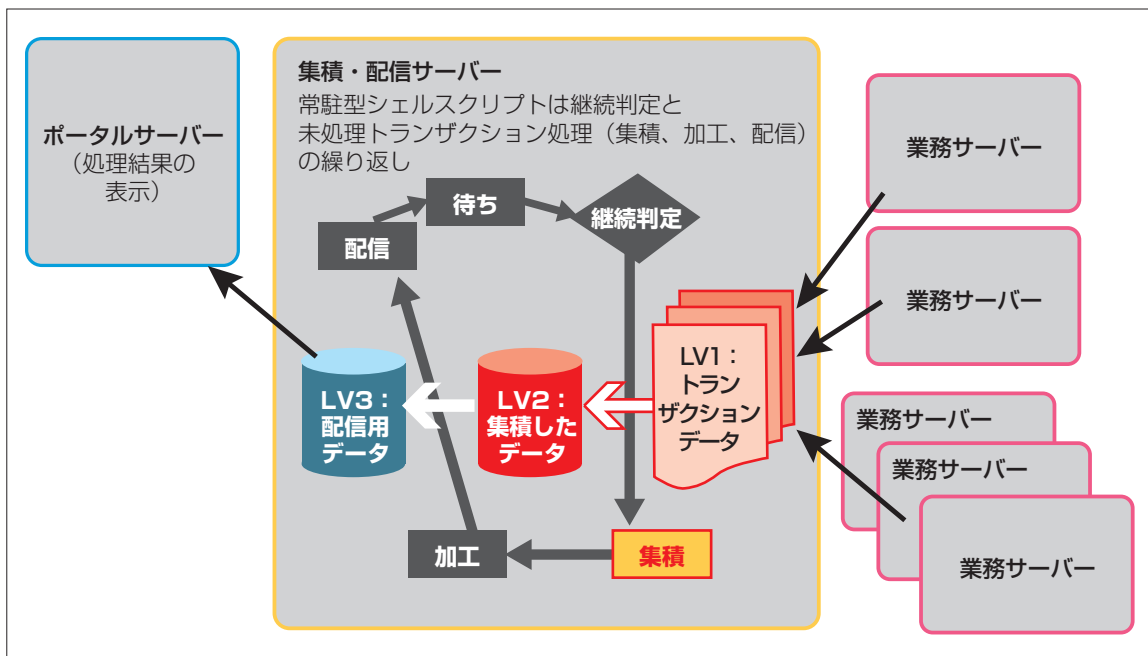
ョンファイルを定期的集め、1ファイルにして保存することです。

技術的な概要

このシェルスクリプトは、発注処理・伝票処理といった各業務から発生するトランザクションデータ、すなわちユニケーij開発手法におけるLV1（システムに入力された生データ）を一定間隔ごとに集積して、LV2 ファイル（LV1 を変換・整形したテキストデータ）として保存する役目を果たします（図2）。

発生するトランザクションには、実データファイルとセマフォファイル（→補足1）の2種類があります。トランザクションを集積する（発生したトランザ

図1 常駐型シェルスクリプトの処理フロー



クションを集める)際の注意点は、同一の実データファイルを重複して集積しない、ということです。

もし、重複してデータを集積すると不正確な処理結果が出力されることになります。そうならないように集積処理が終了すると、元のファイル名（未処理ファイル）をリネームしています。処理済みのセマフォファイルには、ファイル名に ZUMI（済み）という文字を付けて未処理のファイルと区別するのです。リネームはセマフォファイルのみにに対して実施します。セマフォファイルのみをリネームする理由は2つあります。

- セマフォファイルを、未処理（処理済み）の判定に使用しているため
- 実データファイルは LV1（生データ）なのでそのままの状態が残す必要があるため（→補足 2）。

【補足 1】ここで使用するセマフォファイルとは、実データファイルが作成途中に取り込まれることを防ぐために、トリガファイルとして実データファイルが完成した直後に生成されるファイルのことです。実データファイルとセマフォファイルは 1 対 1 で存在します。それは、データ取り込みに過不足がないようにセマフォファイルの件数だけ取り込み処理を実施していることが理由です。実データファイルとセマフォファイルのファイル名は同一にします（画面 3）。

【補足 2】ユニケーシ開発手法では、LV1 データ（生データ）は消去せず保存します。サーバーのディスク容量が不足する場合には、バックアップサーバー等で保持します。プログラムは消去されても後から再生することができず、データは一度失われると二度と再生できないからです。

コードの見どころ

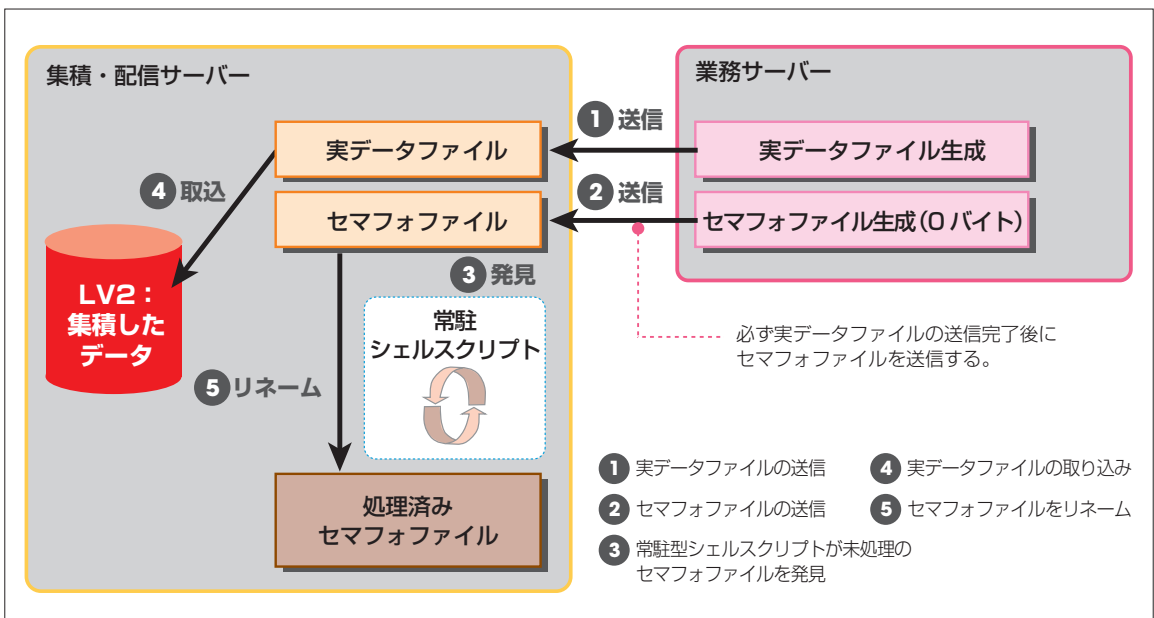
掲載したソースコードは、繰り返し実行される処理（集積・加工・配信・継続判定）の中に含まれている一工程（集積）です（図2）。各トランザクションファイルを集積してファイルをまとめる処理ですが、これは LV1 から LV2 を作成する処理に該当します。その部分を大きく3つのポイントに分けて説明します。

[1] 前回集積した後に発生した新規の未処理ファイル（セマフォファイル）を抽出します。補足1で述べたように、セマフォファイル名は実データファイル名と同一にしているため、セマフォファイル名だけを抜き出して、実データファイルのフルパスを作成します（75～87行目）（→補足 3）。

[2] 重複してデータを取り込まないようにセマフォファイルを処理済みのファイル名にリネームします（98行目）。

[3] 集積したファイルを LV2 として保存します（102～105行目）（→補足4）。

図 2 集積・配信サーバーと業務サーバーの動作



リスト1 集積処理を行うプログラム

```
1 #!/bin/ush -xve
2 # システム名      : U S Pシステム
3 # サブシステム名: 連絡アラートの配信(日次)
4 # 業務名          : 連絡アラート
5 # プログラム名    : 連絡アラートL V 2の集積・通知
6 # 備考(Usage)     : LV2MAKE. TIMES. ALERT_TSUCHI [YYYYMMDD]
7 # シェル名        : LV2MAKE. TIMES. ALERT_TSUCHI
8 # 作成日          : 2013/07/31
9 # 会社名          : USP
10 # 作成者         : A. Asaba
11
12 #/////////////////////////////////////////////////////////////////
13 # 初期設定
14 #/////////////////////////////////////////////////////////////////
15
16 # 走行ログの記録
17 logd="/${HOME}/LOG" # ログディレクトリ
18 logf="/${logd}/LOG. ${basename $0}. ${date +%Y%m%d}_${date +%H%M%S}_$$" # ログファイル名
19 echo "${logf}" &> /dev/null
20 log 2> ${logf}
21
22 # パスの定義
23 PATH=/home/UTL:/home/TOOL:/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin:${PATH}
24 LANG=ja_JP.UTF-8
25
26
27 #-----
28 # 変数の定義
29 #-----
30 tmp="/tmp/${$}-${basename $0}_${date +%Y%m%d%H%M%S}" # 一時ファイル
31 hostname="${HOSTNAME}" # サーバー名
32 semd="/${HOME}/SEMAPHORE" # セマフォディレクトリ
33 subd="ALERT "
34 rcvd=/home/RCV/DATA/ALERT
35 rcvsemd=/home/RCV/SEMAPHORE/ALERT
36
37 lv2d=/home/DATA/LV2
38 outd="/${lv2d}/${subd}"
39
40 # エラー時の終了処理定義 (ushエラーハンドラ)
41 err ERROR_EXIT() {
42     echo "${hostname} ${basename $0}_${sday} ERROR ${date +%Y%m%d%H%M%S} ${logf}" >> ${logd}/UPCNT
43     touch ${semd}/${basename $0}.${hostname}.ERROR.${sday}
44     exit 1
45 }
46
47 # 処理日付の確認
48 sday=${date +%Y%m%d}
49 # 引き数の確認
50 [ $# -eq 1 ] && sday=$1
51
52 # 現在時刻
53 stime="${date +%H%M%S}"
54
55 # 簡易YYYYMMDD日付チェック
56 if ! isdate ${sday} ; then
57     echo "Parameter DATE error:[${sday}]"
58     ERROR_EXIT
59 fi
```

画面1 echo でファイルリストを出力

echoでファイルのリストを出力すると1行に全てが出力される。そこで横型データを縦型に並び替えるコマンドtarrを使用すると、ls同様にデータは縦に並んだ状態で出力される。

```
$ echo /usr/bin/*
/usr/bin/411toppm /usr/bin/BMPtoRGB /usr/bin/DIIP
ugInTester
$ echo /usr/bin/* | tarr
/usr/bin/411toppm
/usr/bin/BMPtoRGB
/usr/bin/DIIPugInTester
lsは外部コマンドなので全件抽出できない場合がある。
$ ls /usr/bin/* | head
/usr/bin/411toppm
/usr/bin/BMPtoRGB
/usr/bin/DIIPugInTester
```

```

60
61 # 前回セマフォの消去
62 rm -f ${semd}/${basename $0}.${hostname}.${sday} >/dev/null 2>&1
63
64 # 起動時刻の記録
65 echo "${hostname} ${basename $0}_${sday} START $(date +%Y%m%d%H%M%S)" >> ${logd}/UPCNT
66 touch ${semd}/${basename $0}.${hostname}.START.${sday}
67
68 #/////////////////////////////////////////////////////////////////
69 # データ処理部
70 #/////////////////////////////////////////////////////////////////
71
72 # 初期化
73 : > ${tmp}-lv2_data
74
75 echo ${rcvsemd}/${sday}/${sday}?????
76 tarr
77 ugrep -v '???'
78 ugrep -v '*'
79 xargs -J % find % -type f
80 xargs basename
81 # ファイルを変換
82 while read fname; do
83   rcvf="${rcvfd}/${sday}/${fname}"
84   [ -e ${rcvf} ] || ERROR_EXIT
85
86 # ファイルの集積
87 cat ${rcvf} >> ${tmp}-lv2_data
88 # 1:アラートメッセージ番号      2:発信者区分      3:発信会社コード
89 # 4:問合せ先                    5:重要度          6:種別
90 # 7:件名パラメータ1            8:件名パラメータ2  9:件名パラメータ3
91 # 10:発信日時                  11:作業期限        12:掲載開始日
92 # 13:受信会社コード           14:受信者ログインID 15:未読フラグ
93 # 16:業務キー1                17:業務キー2       18:業務キー3
94 # 19:業務キー4                20:業務キー5
95 # 21:更新フラグ  22:適用日    23:ユーザID    24:入力日時
96
97 # セマフォを済みとする
98 mv ${rcvsemd}/${sday}/${fname} ${rcvsemd}/${sday}/${fname}.ZUMI
99 done
100
101 ## LV2の集積
102 if [ -s ${tmp}-lv2_data ]; then
103   mkdir -p ${outd}/${sday}
104   cp -p ${tmp}-lv2_data ${outd}/${sday}/${subd}.${sday}${stime}.${sday}
105 fi
106
107 #/////////////////////////////////////////////////////////////////
108 # 終了
109 #/////////////////////////////////////////////////////////////////
110 # 終了時刻の保存
111 echo "${hostname} ${basename $0}_${sday} END $(date +%Y%m%d%H%M%S)" >> ${logd}/UPCNT
112 touch ${semd}/${basename $0}.${hostname}.END.${sday}
113 #ワークファイルの削除
114 set +x
115 rm -f ${tmp}~* >/dev/null 2>&1
116 set -x
117 echo "${basename $0} exit 0"
118 exit 0

```

「コードの見どころ」
および画面2を参照

echo でファイルリス
トを出力 (画面1を参
照)

セマフォファイル (画
面3を参照)

ugrep は grep のラッ
パー。指定パター
が見つからなくてもエ
ラーステータスは「1」
でなく「0」となる。

[1] 未処理ファイルを
抽出して1ファイルに
集積 (75 ~ 87行目)

実ファイル (画面3を
参照)

[2] セマフォファイル
(未処理) を処理済み
に変更 (画面3を参照)

[3] 集積したファイル
をLV2として保存(102
~ 105行目)

「コードの見どころ」
および画面2を参照

画面2 開始・終了を記録するUPCNTファイル

```
$ grep LV2MAKE. TIMES. ALERT_TSUCHI ${logd}/UPCNT
.....
USPSV01 LV2MAKE. TIMES. ALERT_TSUCHI_20131210 START 20131210085500 ← 開始
USPSV01 LV2MAKE. TIMES. ALERT_TSUCHI_20131210 END 20131210085520 ← 終了 } 処理時間20秒
USPSV01 LV2MAKE. TIMES. ALERT_TSUCHI_20131210 START 20131210090020 ← 開始
USPSV01 LV2MAKE. TIMES. ALERT_TSUCHI_20131210 END 20131210090030 ← 終了 } 処理時間10秒
USPSV01 LV2MAKE. TIMES. ALERT_TSUCHI_20131210 START 20131210090535 ← 開始
USPSV01 LV2MAKE. TIMES. ALERT_TSUCHI_20131210 END 20131210090552 ← 終了 } 処理時間17秒
.....
```

画面3 セマフォファイルの生成

```
$ ls -l /home/RCV/DATA/ALERT/20131210/ALART_MSG. END. 20131210090000
-rw-rw-r-- 1 usp usp 1120 12月 10日 /home/RCV/DATA/ALERT/20131210/ALART_MSG. END. 20131210090000 ◆----- 実ファイル

$ ls -l /home/RCV/SEMAPHORE/ALERT/20131210/ALART_MSG. END. 20131210090000
-rw-rw-r-- 1 usp usp 0 12月 10日 /home/RCV/SEMAPHORE/ALERT/20131210/ALART_MSG. END. 20131210090000 ◆----- セマフォファイル

取り込み処理が終了するとセマフォファイルをZUMI付きに変更

$ ls -l /home/RCV/SEMAPHORE/ALERT/20131210/ALART_MSG. END. 20131210090000. ZUMI
-rw-rw-r-- 1 usp usp 0 12月 10日 /home/RCV/SEMAPHORE/ALERT/20131210/ALART_MSG. END. 20131210090000. ZUMI
```

各業務サーバーからのトランザクション（実ファイル）が生成された後に、0バイトのセマフォファイルが生成される

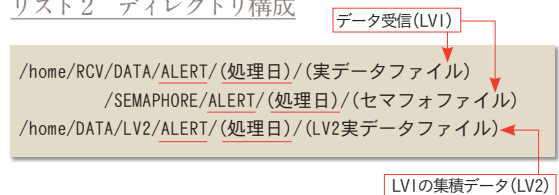
ユニケーj開発手法においては、開始・終了を記録するUPCNTファイルを用意します（65、111行目）。UPCNTファイルには、サーバー名・シェルスクリプト名・処理日・ステータス（開始／終了／エラー）・日時などの有用なデータを記録します。これらの情報から、シェルスクリプトの処理時間やCGIの場合には時間常別アクセス回数などを算出することができます（画面2）。

さらにディレクトリ内にあるファイルを抽出して、ファイルリストを作成します（75行目）。ファイルを抽出する代表的なコマンドにecho、find、lsがあります。抽出する条件に「*」や「?」などを付けて実行した場合、echoはシェル（bash、ushなど）の内部コマンドであることから、件数が多くてもデータを抽出することができます。一方、findとlsは外部コマンドであるため件数が多すぎると「引数のリストが長すぎます」といったメッセージが出てすべて抽出できない場合があります。こうした違いを考慮して使い分ける必要があります。echoコマンドの使用例は画面1（P.28）をご覧ください。

【補足3】 今回のケースでは複数のトランザクションファイルを1ファイルにまとめましたが、1ファイルに集積できない場合、1トランザクションを1ファイル（LV2）として保存する場合もあります。

【補足4】 ユニケーj開発手法では、入力業務の種類や時系列（日時など）ごとにディレクトリを分けてデータ配置します。以下は、システムのディレクトリ構成の例です。

リスト2 ディレクトリ構成



まとめ

セマフォファイルを上手に扱うことで、正しく実データを集積することができます。今回のようにセマフォファイル（トリガファイル）を使用しないで、実データファイルを作成途中で集積すると、中途半端な状態で取り込んでしまい、後々トラブルの種になる場合があります。

さて、今回は、配信用LV3データを他サーバーへ配信する処理について説明します。