



Nutzung von Linuxbefehlen unter SAS

- ▶ KSFE-Vortrag, Donnerstag 15.9.2022, 16:30 h, ADMINISTRATION

KSFE 2022

Thomas Rüdiger, thomas.ruediger@axa.de

Volle Fahrt voraus mit SAS auf Linux! Warum ist das so beliebt?

„Dem Data-Warehouse- und Business-Intelligence-Spezialisten zufolge haben in einer kürzlich durchgeführten Umfrage 90 Prozent der befragten Unternehmen den Wunsch geäußert, das "SAS System" auf Linux-Plattformen einzusetzen.“
(03.04.2000, COMPUTERWOCHE)

“Unix SAS installations are done on RedHat Enterprise Linux (RHEL) these days. Linux has lower cost of ownership, no real memory limitations, and lighter-weight OS background services than other operating systems, such as Windows.” (19.09.2020, zencos, Andy Peredery)



Linuxbefehle über SAS starten Bourne-Shell & Co



Linuxbefehle weisen Ähnlichkeiten mit DOS-Befehlen auf, bspw. Datei-Kopieren (Linux: cp DOS: copy, Verschieben (Linux: mv, DOS: move).

<https://archive.download.redhat.com/pub/redhat/linux/7.3/en/doc/RH-DOCS/rhl-gsg-en-7.3/ch-doslinux.html>

Für ausführbare Datei-Operationen lässt der SAS-system-Befehl verwenden:

```
data _null_;  
  Linuxbefehl= "chmod 0540 $HOME/*.csv" ;  
  RC=system(Linuxbefehl) ;  
  if RC^=0 then put "ERROR:" Linuxbefehl= RC= ;  
run;
```

Linux-Bildschirmausgaben lassen sich per filename-pipe-Statement einlesen.

```
filename PIPE pipe "ls -l --time-style=full-iso $HOME";
```

```
data _null_;  
  infile PIPE lrecl=32767 ;  
  input ;  
  put _infile_ ;  
run;
```

A screenshot of an MS-DOS command prompt is shown in the bottom right. The text 'Starting MS-DOS...' is at the top, and the prompt 'C:\>_' is at the bottom, indicating the start of a command line session.

```
Starting MS-DOS...  
C:\>_
```

Linux als SAS-System-Helfershelfer

Einige ausgewählte Fragen

1. Wieviel Speicherplatz haben wir noch in unseren Linux-Verzeichnissen?
2. Welche SAS-Tabellen sind zu groß oder „ungenutzt“ veraltet?
3. Kann ich SAS-Tabellen per Linux-Befehl komprimieren/archivieren?
4. Kann ich SAS-WORK-Tabellen per Linux allein durch Verschieben dauerhaft speichern?
5. Kann ich SAS-Jobs mit versehentlicher Endlosschleife per Linux stoppen?
6. Welche Berechtigungen und SAS-Rollen habe ich und was haben andere, was ich nicht habe?
7. Wie lange ist mein Linux-Passwort noch gültig?

Linuxinformationen aus SAS-automatischen Macrovariablen

Über die automatischen SAS-Macrovariablen (&sys...) erhält man

- Hostinformationen (entspricht Linuxbefehl `uname`)

```
%put #&=SYSSCPL.#&=SYSSCP.#&=SYSHOSTINFOLONG.;
```

```
%put #&=SYSTCPIPHOSTNAME.#&=SYSHOSTNAME.;
```

- Linux-Prozess-ID zum SAS-Enterprise Guide (wiederzufinden über Linuxbefehl `ps`)

```
%put &=SYSJOBID.;
```

- Linux-Prozess-ID zum SAS-Enterprise Guide (entspricht Linux-Umgebungsvariable `$USER`)

```
%put &=SYSUSERID.;
```

- Die Linux-Umgebungsvariable `$HOME` entspricht i.d.R. dem Verzeichnis zur SASUSER-Umgebung `/home/<Userid>`

1. Speicherplatz pro Filesystem Wann wird es eng?



Der df-Befehl gibt Auskunft über den Speicherplatz.

```
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/mapper/&syshostname._DATA-sashome
```

```
filename PIPE pipe "df -v";
```

```
data Disk_Available(keep=Verzeichnis--Available);
infile PIPE lrecl=32767;
input;length Verzeichnis $40 Filesystem $50 Used_Prozent Used_Blocks_Prozent blocks_1024 Used Available 8 Pos _I_ 3 Text
$200;
format blocks_1024 Used Available commax32. Used_Prozent Used_Blocks_Prozent commax6.1;retain Filesystem;
if _infile_ ^= '' and scan(_infile_,1,byte(32)) ^= 'Filesystem';infile_=left(_infile_);
if substr(_infile_,1,1) in ('/' 't') then do;
  Pos=index(_infile_,byte(32));
  if Pos>1 then do;Filesystem=left(substr(_infile_,1,Pos-1));_infile_=left(substr(_infile_,Pos+1));end;
  else Filesystem=_infile_;
end;
if _infile_ ^= '' and _infile_ ^= Filesystem then do;
  do _I_=1 to 4;Pos=index(_infile_,byte(32));
  if Pos>1 then do;
    Text=left(substr(_infile_,1,Pos-1));
    select (_I_);
      when (1) blocks_1024=input(Text,best32.);
      when (2) Used=input(Text,best32.);
      when (3) Available=input(Text,best32.);
      when (4) Used_Prozent=input(translate(Text,','','%'),best32.);
      otherwise put "ERROR:" _I_=;
    end;
    _infile_=left(substr(_infile_,Pos+1));
  end;
end;
Verzeichnis=_infile_;
Used_Blocks_Prozent=100*Used/blocks_1024;
output;end;
```

run;

Verzeichnis	Filesystem	Used_Prozent	blocks_1024	Used	Available
/home	/dev/mapper/&syshostname._DATA-sashome	67,00%	51.606.140	32.578.120	16.407.172

2. Dateigröße und Dateialter Zu groß? Zu alt?



Der ls-Befehl listet Verzeichnis- und Dateiinformationen.

```
-rw-r--r-- 1 sas sas 39333888 2018-10-25 04:24:14.000000000 +0200 /opt/sas/sas_94/SASFoundation/9.4/sashelp/draft.sas7bcac
```

```
filename PIPE pipe "ls -l --time-style=full-iso /opt/sas/sas_94/SASFoundation/9.4/sashelp/*";

data Dateien(keep=Datei--Datum_Zeit);
infile PIPE lrecl=32767;
input;_infile_=left(tranwrd(_infile_,"/opt/sas/sas_94/SASFoundation/9.4/sashelp/",""));
length Datei $200
       Berechtigung_Owner Berechtigung_Gruppe Berechtigung_Andere $3
       Datei_Owner $12 Owner_Gruppe $12 Bytes 8 Datum_Zeit 8 Datum_Zeit_Text $19
       Verzeichnis_Datei $1;

format Bytes comma32. Datum_Zeit deufdfdt19.;
Verzeichnis_Datei=translate(translate(substr(_infile_,1,1),'V','d'),'D','-');
if Verzeichnis_Datei='D' then do;
  Berechtigung_Owner=substr(_infile_,2,3);
  Berechtigung_Gruppe=substr(_infile_,5,3);
  Berechtigung_Andere=substr(_infile_,8,3);
  Datei_Owner=scan(_infile_,3,byte(32));
  Owner_Gruppe=scan(_infile_,4,byte(32));
  Bytes=input(compress(scan(_infile_,5,byte(32)),'0123456789','k'),best32.);
  Datum_Zeit_Text=scan(_infile_,6,byte(32))!!byte(32)!!scan(scan(_infile_,7,byte(32)),1,'. ');
  Datum_Zeit=input(Datum_Zeit_Text,ydddtm19.);
  Datei=left(substr(_infile_,index(_infile_,trim(Datum_Zeit_Text))
    +length(Datum_Zeit_Text)
    +length(scan(scan(_infile_,7,byte(32)),2,'. '))
    +length(scan(_infile_,8,byte(32)))
    +2
  ));
output;
end;

run;
proc sort noequal nodupkey data=Dateien;by descending Bytes Datei;run;
```

Datei	Berechtigung _Owner	Berechtigung _Gruppe	Berechtigung _Andere	Datei_ Owner_	Owner Gruppe	Bytes	Datum_Zeit
draft.sas7bcac	rw-	r--	r--	sas	sas	39.333.888	25Okt2018:4:24:14

3. Uralte „ungenutzte“ Dateien Ab damit aufs Abstellgleis!



Der gzip-Befehl komprimiert SAS-Tabellen und macht sie wegen der Dateiendung .gz in SAS unsichtbar. `gzip -d *.gz` macht die SAS-Tabellen wieder sichtbar.

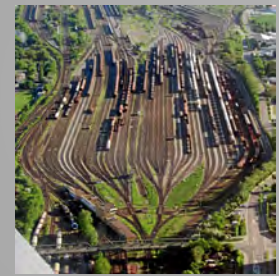
```
data
  zipcode_no      (compress=no)
  zipcode_binary (compress=binary)
  zipcode_char    (compress=char);
set SASHELP.zipcode;

run;
data _null_;
  Statement="gzip"!!byte(32)
  !!trim(pathname("WORK"))!!"/zipcode*.sas7bdat";
  RC=system(Statement);
  if RC^=0 then put "ERROR:" Statement RC=;

run;
```

SAS Tabelle unkomprimiert	Größe	SAS Tabelle komprimiert	Größe	Reduktion um
zipcode_binary.sas7bdat	6.881.280	zipcode_binary.sas7bdat.gz	2.332.241	66,11%
zipcode_char.sas7bdat	6.750.208	zipcode_char.sas7bdat.gz	2.025.398	70,00%
zipcode_no.sas7bdat	36.200.448	zipcode_no.sas7bdat.gz	2.092.146	94,22%

4. WORK-Tabellen auf Dauer verschieben



Der mv-Befehl kann SAS-Tabellen unverändert verschieben und ist in der Regel bei Verzeichniswechsel schneller als Tabellenkopieren in SAS.

```
data _null_;
  if exist("zipcode_char")^=0 then do;
    Statement="mv"!!byte(32)!!trim(pathname("WORK"))
    !!"/zipcode_char.* $HOME/.";
    RC=system(Statement);
    if RC^=0 then put "ERROR:" Statement= RC=;
  end;
  else put "WARNING: Datei zipcode_char existiert nicht!";
run;
```

5. SAS-Linux-Prozesse killen

Endlosem einen kurzen Prozess machen



Eigene lästige SAS-Prozesse (z. B. aus Endlos-Schleifen) lassen sich mit dem ps-Befehl erkennen und mit dem kill-Befehl beenden.

```
<Userid> 54300 130939 0 13:38 ? 00:00:01 /opt/sas/sas_94/SASFoundation/9.4/sasexe/sas ...
```

```
filename PIPE pipe 'ps -ef | grep -v grep | grep -v "ps -ef" | grep "/sas" | grep $USER';
data Linux_Prozesse(keep=User_Id--Linux_Command);
  infile PIPE lrecl=32767;input;
  length User_Id $20 Process_Id 8 Parent_Process_Id 8 CPU_Percent 8 Start 8 Terminal $5
         CPU_Time 8 Linux_Command $1200 Pos _I_ 3 Text $80;
  format Start deufdfd19. CPU_Time time8.;informat Start datetime19. CPU_Time time8.;
  if _infile_ ^= ' ' and scan(_infile_,1,byte(32)) ^= 'UID';
  do _I_=1 to 7;
    Pos=index(_infile_,byte(32));
    if Pos>1 then do;
      select (_I_);
        when (1) User_Id=left(substr(_infile_,1,Pos-1));
        when (2) Process_Id=input(left(substr(_infile_,1,Pos-1)),best32.);
        when (3) Parent_Process_Id=input(left(substr(_infile_,1,Pos-1)),best32.);
        when (4) CPU_Percent=input(left(substr(_infile_,1,Pos-1)),best32.);
        when (5) do;
          Text=left(substr(_infile_,1,Pos-1));
          if index(Text,':')^=0 then Start=dhms(date(),input(scan(Text,1,':'),best.),input(scan(Text,2,':'),best.),0);
          else do;
            if length(Text)=5 then Text=substr(Text,4,2)!!upcase(substr(Text,1,3))!!put(date(),year4.);
            Start=dhms(input(Text,date9.),0,0,0);
          end;
        end;
        when (6) Terminal=left(substr(_infile_,1,Pos-1));
        when (7) CPU_Time=input(left(substr(_infile_,1,Pos-1)),time8.);
        otherwise put "ERROR:" _I_=;
      end;_infile_=left(substr(_infile_,Pos+1));
    end;
  end;Linux_Command=_infile_;
run;
```

kill -9 54300

User_Id	Process_Id	Parent_Process_Id	CPU_Percent	Start	Terminal	CPU_Time	Linux_Command
<Userid>	54300	2254	0	11Sep2022:18:14:00?		00:00:04	.../sasexe/sas ...

6. Berechtigungen Haben oder nicht haben



Der `getent group`-Befehl gibt sämtliche eigene Linux-Berechtigungen (z.B. SAS Rollen) und die anderer User wieder.

```
%let Userid=&sysuserid.*&sysuserid.|$USER|andere Userid;
filename PIPE pipe "getent group | grep &Userid. | cut -f1 -d':' | sort";
data Meine_Linux_Rollen;
  length Userid $7 Linux_Rolle $32;
  retain Userid "&Userid.";
  infile PIPE lrecl=32767;
  input;
  Linux_Rolle=_infile_;
run;
filename PIPE clear;
```

Userid	Linux_Rolle
<Userid>	SAS-ROLLE_01
<Userid>	SAS-ROLLE_99
<Userid>	ORGANISATIONSEINHEIT

7. Linux-Passwort Wann muss ich es ändern?



Viele persönliche Informationen befinden sich in LDAP (**L**ightweight **D**irectory **A**ccess **P**rotocol) und lassen sich mit dem `ldapsearch`-Befehl auslesen.

```
filename PIPE pipe "ldapsearch -x uid=$USER uid gecos gidNumber shadowLastChange shadowMax";
data ldap_ein_user(keep=Verbindung_kurz--Password_Tage);
length Verbindung_kurz $64 Verbindung $74 Gruppe 8 Userid $20 Username $36 Password_Aenderung 5 Password_Tage 3 Text $80;
retain Verbindung_kurz Verbindung Gruppe Userid Username Password_Aenderung Password_Tage;
format Password_Aenderung deufdd10.;
informat Password_Aenderung ddmmyy10.;
infile PIPE lrecl=32767;
input;retain Step '0';if index(_infile_,'requesting')^=0 then Step='1';if Step='1';
if _infile_not in (' '#') and index(_infile_,'requesting')=0
and index(_infile_,'search:')=0 and index(_infile_,'result:')=0
and index(_infile_,'search result')=0 and index(_infile_,'numResponses:')=0
and index(_infile_,'numEntries:')=0
;_infile_=left(_infile_);
Text=_infile_;
if substr(Text,1,1)='#' then Verbindung_kurz=left(substr(Text,2));
else if substr(Text,1,3)='dn:' then Verbindung=left(substr(Text,4));
else if substr(Text,1,4)='uid:' then Userid=left(substr(Text,5));
else if substr(Text,1,6)='gecos:' then Username=left(substr(Text,7));
else if substr(Text,1,10)='gidNumber:' then Gruppe=input(compress(substr(Text,11)),best32.);
else if substr(Text,1,17)='shadowLastChange:' then Password_Aenderung=input(left(substr(Text,18)),best32.)+3653;
else if substr(Text,1,10)='shadowMax:' then Password_Tage=input(left(substr(Text,11)),best.);
output;
run;
data ldap_ein_user;
set ldap_ein_user end=EOF;
if EOF;
Password_Aenderung=Password_Aenderung+Password_Tage;
run;
```

Gruppe	Userid	Username	Password_Aenderung	Password_Tage
<Gruppe>	<Userid>	RUEDIGER,THOMAS	20.09.2022	40

Vielen Dank für Ihre/Eure Aufmerksamkeit
und bis zur nächsten KSFE!



Thomas Rüdiger, thomas.ruediger@axa.de