Przykładowy skrypt

Skrypt korzysta ze zmiennych środowiskowych, konstrukcji decyzyjnych oraz zaawansowanych funkcji shella do wykasowania wszystkich plików należących do aktualnego użytkownika i znajdujących się w katalogu /tmp. Operacja ta jest często przeprowadzana przez administratorów systemu, ponieważ pewne programy nie sprzątają po sobie, a niektórzy użytkownicy lubią wykorzystywać katalog /tmp do przechowywania własnych danych. Specjalną funkcją skryptu, udostępnianą wyłącznie administratorom, jest możliwość uzupełnienia jego nazwy o nazwę użytkownika, którego pliki mają zostać wykasowane.

Skrypt czysctmp służy do usuwania z katalogu /tmp plików należących do aktualnego użytkownika (lub do użytkownika o nazwie podanej jako parametr przez administratora)

```
1.  #!/bin/bash
2.  #czysctemp
3.  # skrypt usuwajacy z katalogu /tmp wszystkie pliki nalezace do aktualnego uzytkownika
4.  #wywolanie: czysctmp <uzytkownik>
5.  #argument <uzytkownik> jest dostepny tylko dla administratora (root)
6.
7.  # sprawdzamy, czy skrypt zostal uruchomiony przez roota
8.  if [$LOGNAME = 'root']; then
9.      if [$1]; then
10.         NAZWA=$1
11.       else
12.         NAZWA=$LOGNAME
13.     fi
14.  else
15.     if [$1]; then
16.         echo "Tylko administrator moze kasowac pliki nalezace do innych uzytkownikow."
17.         exit 1
18.       else
19.         NAZWA=$LOGNAME
20.     fi
21. fi
22. echo "Kasujemy pliki uzytkownika "$NAZWA
23. echo "Wykonanie tej operacji może spowodowac problemy, jeśli kasowane pliki"
24. echo "sa wykorzystywane przez inne aplikacje."
25. echo -n "Czy chcesz skasowac pliki? [t/N]"
26. read
27. if [$REPLY]; then
28.     if [$REPLY='t' -o $REPLY='T']; then
29.         echo "Potwierdziles chec skasowania plikow."
30.       else
31.         echo "Nie potwierdziles che ci skasowania plikow."
32.         exit 0
33.     fi
34.   else
35.       echo "Nic nie odpowiedziales."
36.       exit 0
37.  fi
38. #co sprowadza nas do...
39. echo -n "Kasujemy pliki nalezace do uzytkownika "$NAZWA" i umieszczone w katalogu
    /tmp...".
40. rm -f $(find /tmp -user $NAZWA)
41. echo "gotowe."
42. # koniec skryptu
```

Oto opis ciekawszych fragmentów skryptu:

- W linijce 8 wprowadzamy polecenie if razem z wyrażeniem testowym oraz odwołujemy się do pierwszej zmiennej, która jest tak naprawdę jedną ze zmiennych środowiskowych zwracanych przez env i zawiera nazwę aktualnego użytkownika. Wyrażenie if umożliwia wykonanie wszystkich poleceń umieszczonych po then wtedy i tylko wtedy, gdy warunek

następujący po if jest spełniony (lub posiada wartość l). W tym przypadku porównujemy wartość zmiennej LOGNAME oraz łańcuch root, aby określić, czy użytkownik, który uruchomił skrypt jest administratorem.

- W linijce 9 ponownie korzystamy ze składni if do sprawdzenia, czy nazwę skryptu uzupełniono o argument w wierszu poleceń. Do argumentów wpisanych w wierszu poleceń odwołujemy się podając ich numer: $1 to wartość pierwszego argumentu, $2 drugiego i tak dalej. Jeśli potrzebny nam jest łańcuch złożony ze wszystkich argumentów, należy użyć zmiennej $@. Umieszczenie nazwy zmiennej w wyrażeniu testowym (po if) powoduje wykonanie poleceń wypisanych po then wtedy i tylko wtedy, gdy podana zmienna zawiera wartość (jest niepusta).
- W linijce 10 przypisujemy wartość nowej zmiennej NAZWA. Należy zauważyć, że znak równości i wartość zmiennej nie są rozdzielone przez żaden odstęp. Jeśli przypisywana zmiennej wartość zawiera odstępy, należy ją wziąć całą w apostrofy.
- Linijki 8-21 służą do zweryfikowania poprawności podanych argumentów w następujący sposób:

```
jeśli użytkownik jest administratorem (początek wyrażenia if wówczas
jeśli wykryto argument (początek wyrażenia if)
  przypisz wartość tego argumentu zmiennej NAZWA
w przeciwnym wypadku
    przypisz zmiennej NAZWA wartość root
 koniec wyrażenia if
w przeciwnym wypadku
    jeśli wykryto argument (początek. wyrażenia if)
    poinformuj użytkownika o błędzie i zakończ
  w przeciwnym wypadku
      przypisz zmiennej NAZWA nazwę aktualnego użytkownika
 koniec wyrażenia if
koniec wyrażenia if
```

Należy pamiętać, że w przeciwieństwie do większości języków programowania, w shellu bash wyrażenia if kończy się wpisując fi (nie endif ani end).

- Linijki 22-24 ostrzegają użytkownika przed potencjalnie niepożądanymi skutkami przeprowadzanej operacji. Umieszczanie takiego ostrzeżenia w programach modyfikujących dane na dysku jest bardzo ważne; w przeciwnym wypadku można jako administrator systemu stać się adresatem listów, w których użytkownicy będą narzekać na utratę danych.
- Linijka 25 wyświetla znak gotowości, a opcja -n polecenia echo powoduje pozostawienie kursora w miejscu, w którym zakończono wyświetlanie tekstu, aby zasygnalizować, że program oczekuje na odpowiedź.
- Linijka 26 zawiera polecenie read Powoduje ono wczytanie pojedynczego wiersza tekstu z klawiatury i umieszcza każde słowo w następujących po read zmiennych (w kolejności, w jakiej zostały one wypisane). Do ostatniej z podanych zmiennych zostaną wpisane wszystkie wyrazy, które nie zmieściły się w poprzednich. Jeśli nie podamy żadnych nazw zmiennych, wówczas cały tekst zostanie umieszczony w zmiennej REPLY.
- Linijki 28-38 sprawdzają co zawiera zmienna REPLY (jeśli nie zawiera nic, oznacza to, że użytkownik wcisnął po prostu Enter). Jeśli znajdziemy wartość, sprawdzamy, czy jest ona równa "t" (względnie "T"). Jest to jedyna sytuacja, w której skrypt skasuje pliki; wszystkie inne odpowiedzi spowodują natychmiastowe zakończenie jego działania (z jednoczesnym poinformowaniem użytkownika, dlaczego tak się stało).
- Linijka 40 stanowi punkt dojścia - uruchamia polecenie rm -f dla wszystkich plików, których właścicielem jest użytkownik o nazwie przechowywanej w zmiennej NAZWA. Na koniec wyświetlany jest jeszcze jeden komunikat Tym razem mówiący o zakończeniu kasowania plików. Należy zauważyć, że polecenie find, umieszczone w nawiasach po $, zostaje wykonane w pierwszej kolejności, a jego wynik przekazany do
  rm -f.