

מערכות פתוחות

חיים צדוק
קיריל פליסקו
גריגייל בע"מ

מערכות פתוחות

בהתאם לתכנית הלימודים
מגמת תקשוב, ת"ל מערכות פתוחות

מאת
חיים צדוק
קיריל פליסקו
גריגייל בע"מ

Copyright © 2010 by Grigale LTD
כל הזכויות שמורות לגריגייל בע"מ 2010

אין לראות בספר זה ובתכניו מידע מייצג עבור ת"ל
מערכות פתוחות

תוכן עניינים

3.....	תוכן עניינים.....
6.....	כרך 2.....
6.....	ניהול תחנת קצה.....
7.....	ניהול תחנת קצה סולריס.....
8.....	ניהול התקני חומרה.....
9.....	שימוש בכלי ההתקנים.....
9.....	עץ החומרה.....
9.....	שם התקן פיזי.....
9.....	שם התקן לוגי.....
9.....	שם התקן מופע - Instance name.....
10.....	סוגי התקנים.....
10.....	התקן מסוג character.....
11.....	התקן מסוג block.....
13.....	ניהול דיסקים.....
14.....	MBR – טבלת המחיצות.....
15.....	LABEL/VTOC – טבלת המחיצות.....
16.....	מיקום התקני דיסקים.....
16.....	מוסכמת שם לדיסקים.....
17.....	Storage Multipathing.....
17.....	הפקודה fdisk.....
18.....	הפקודה format.....
19.....	הפקודה rmformat.....
20.....	התקנת חבילות תוכנה וטלאי מערכת הפעלה.....
21.....	ניהול חבילות.....
21.....	SVR4 packages.....
21.....	IPS.....
21.....	ניהול טלאים.....
22.....	כלי מנהל חבילות.....
22.....	נעילת מסד נתוני החבילות.....
22.....	ניהול סביבות ריצה.....
23.....	תרגילי בית.....
24.....	ניהול מערכת הקבצים ZFS.....
25.....	הסבר על מבנה ZFS.....
26.....	תכונות של ZFS.....
27.....	מערכת קבצים טרנזקציונית.....
28.....	התאוששות עצמית.....
29.....	התאוששות עצמית למידע יתיר.....
30.....	ניהול מאגרים.....
30.....	הרצת סימולציה.....
30.....	יצירת מאגר מסוג Stripe.....
31.....	יצירת מאגר מסוג mirror.....
31.....	המרת מאגר מסוג Stripe למאגר מסוג Mirror.....

31.....	המרת מאגר מסוג mirror למאגר רגיל ללא יתירות.
32.....	יצירת מאגר מסוג RAIDZ1 (RAID5).
32.....	יצירת מאגר מסוג RAIDZ2.
32.....	הריסת מאגר.
33.....	כיצד לצפות בסטטוס של מאגר.
33.....	צפיה במידע על המאגר.
33.....	צפיה בסטטיסטיקות על המאגר.
33.....	בדיקת תקינות מאגר.
34.....	כיצד לייבא וליצא מאגרים.
34.....	ייצוא של מאגר(התנתקות ממאגר).
34.....	יבוא של מאגר (התחברות למאגר).
36.....	ניהול מערכות קבצים.
36.....	כיצד ליצור ולהרוס מערכות קבצים.
36.....	יצירת מערכת קבצים.
36.....	שינוי שם למערכת קבצים.
37.....	הריסת מערכות קבצים.
37.....	קבלת מידע על מערכת קבצים.
37.....	כיצד לצפות ולשנות תכונות של מערכת קבצים.
38.....	מכסות ושמירת מקום - quotas and reservations.
40.....	כיצד ליצור ולהרוס snapshots.
40.....	יצירת snapshot.
40.....	הריסת snapshot.
40.....	גלגול ל - snapshot קודם.
41.....	יצירה והריסת שיבוטים clones.
41.....	קידום של - clone.
43.....	עבודה עם קבצים כדיסקים.
44.....	רצף האתחול.
45.....	תהליך האתחול במחשב - POST.
46.....	שלב האתחול.
46.....	שלב ה- GRUB.
46.....	שלב הגרעין והשירותים.
47.....	מהו GRUB.
47.....	שלב stage1.
47.....	שלב stage2.
47.....	קובץ menu.lst.
47.....	אפשרויות מתקדמות.
48.....	עליה אינטראקטיבית.
48.....	עליה למצב תחזוקה.
48.....	הפקודה bootadm.
50.....	התאוששות מהגדרות משובשות.
51.....	ניהול שירותי מערכת הפעלה.
52.....	מנהל השירותים.
52.....	ניהול שירות בממשק הפקודה.
52.....	הפקודה sves.
54.....	הגדרת משתמשים וקבוצות.
55.....	שימוש בחלון המשתמשים והקבוצות.

55.....קבצי הגדרות לניהול משתמשים וקבוצות

56.....שימוש בשורת הפקודה

56.....ניהול קבוצות

56.....ניהול משתמשים

57.....צפיה בהגדרות של משתמשים

58.....אבטחה מקומית למערכת ההפעלה

61.....הצפנה או ערבול (Hashing or Encryption)

61.....שימושים

62.....תוכנות מעניינות לשימוש מקומי

כרך 2

ניהול תחנת קצה

על פי מודל סולריס

פרקי הלימוד

ניהול התקני חומרה

ניהול דיסקים

התקנת חבילות תוכנה וטלאי מערכת הפעלה

ניהול מערכת קבצים ZFS

רצף האתחול

ניהול שירותי מערכת הפעלה

משתמשים וקבוצות

אבטחה מקומית למערכת ההפעלה

ניהול תחנת קצה סולריס

מטרות

מטרת הפרק היא ללמד כיצד לנהל סולריס כמערכת בודדת. כיצד לנהל התקני חומרה, חבילות תוכנה וטלאי תוכנה, כיצד לנהל את מערכת הקבצים ZFS, כיצד לנהל את שירותי מערכת ההפעלה, כיצד לנהל משתמשים וקבוצות וכיצד לאבטח את מערכת ההפעלה.

ניהול התקני חומרה

נושאי לימוד:

- שימוש במנהל ההתקנים
- עץ החומרה
- הסבר על סוגי שמות התקנים בסולריס
 - שם פיזי
 - שם לוגי
 - שם הדגם

בפרק זה ילמדו התכנים הבאים:
כיצד מנוהלים התקני חומרה והתקנים נתיקים בסולריס. התקני החומרה בסולריס מנוהלים על ידי היררכיית תיקיות החומרה. התלמיד ידע מהם סוגי שמות התקנים בסולריס: שם פיזי, שם לוגי, שם הדגם (instance), וכיצד להשתמש במנהל ההתקנים החלונאי או ועל ידי שימוש בפקודות `volcheck` ו-`rmformat`.

שימוש בכלי ההתקנים

Applications → System Tools → Device driver utility

ניתן לראות את כל ההתקנים במערכת.
ההתקנים שלא מזוהים יופיעו באדום.
ניתן לבצע רענון לעץ ההתקנים על ידי שימוש בכפתור - Refresh.
ניתן לראות האם יש חומרה לא מזוהה בצד שמאל של החלון למעלה.

עץ החומרה

עץ החומרה בסולריס הנו מערכת קבצים שהספריית שורש שלה היא /devices
ה-BIOS מזהה את החומרה במחשב דרך תהליך ה-POST ומייצר את עץ קבצי החומרה
ומעביר את המידע לגרעין (kernel)

ישנם שלושה סוגי שמות התקנים בסולריס

- שם התקן פיזי
- שם התקן לוגי
- שם התקן מופע

שם התקן פיזי

זהו הנתוב המלא של ההתקן כפי שהוא מופיע ב - /devices

שם התקן לוגי

זהו התקן שהשם שלו אינפורמטיבי וקל לפענוח
כל קבצי ההתקן הלוגיים נמצאים תחת הספרייה /dev והם בדרך כלל קיצורי דרך (symbolic link)
לשמות התקן פיזיים.
והשם שלהם הוא כדוגמת -

/dev/audio/0

שם התקן מופע - Instance name

זהו שם התקן בו גרעין מערכת ההפעלה משתמש כאשר הוא פונה אל אותו התקן.
בדרך כלל שם זה הנו שם המודול גרעין (kernel module) במערכת והוא מורכב מכמה אותיות

ומספר המייצג את המופע שלו במערכת.
כדי לראות את כל המודולים במערכת משתמשים בפקודה:

modinfo
modinfo | grep e1000g

לדוגמא:

e1000g0 – שם התקן מופע לכרטיס רשת הראשון במערכת.
e1000g1 – שם התקן מופע לכרטיס רשת השני מאותו סוג במערכת.

לדוגמא:

שם התקן לוגי audio מציין איזה כרטיס הנו כרטיס קול ברירת מחדל במערכת -

```
haim@os1:~$ ls -l /dev/audio
lrwxrwxrwx 1 root root 7 2010-07-14 11:30 /dev/audio -> sound/0
```

שם התקן זה ממופה ל – sound/0 שממופה ל audio810:0

```
haim@os1:~$ ls -l /dev/sound/0
lrwxrwxrwx 1 root root 10 2010-07-14 11:30 /dev/sound/0 -> audio810:0
```

שם התקן לוגי audio810:0 ממופה לשם התקן פיזי

```
haim@os1:~$ ls -l /dev/sound/audio810:0
lrwxrwxrwx 1 root root 46 2010-07-14 11:30 /dev/sound/audio810:0 ->
../devices/pci@0,0/pci8086,0@5:sound,audio0
```

כדי לעבור משם התקן לוגי לשם מופע ניתן לבצע את הפקודה:
האופציה -L מציגה את הקובץ האחרון בשרשרת הלינקים.

ls -lL /dev/audio

```
crw----- 1 haim staff 112, 3 2010-07-14 11:48 /dev/audio
```

112 - Major/Module number – מספר מודול גרעין.

כדי לראות מי המודול שמספרו 112 יש לבצע:

modinfo |grep 112

בין התוצאות שנקבל, רק השורה שעמודה רביעית בה מכילה 112 היא השורה הרצויה.

3 – minor number – מספר המופע במערכת.

סוגי התקנים

התקן מסוג character

זהו התקן שמדווח לגרעין כי צורת התקשורת מולו הנה ב- characters (תו אחרי תו).

התקן מסוג block

זהו התקן שמדווח לגרעין כי צורת התקשורת מולו הנה ב- Buffer (חוצץ).

דוגמאות להתקנים משני הסוגים:

התקן שעובדים מולו בחוצצים -

```
brw-r----- 1 root sys 262, 16 2010-07-21 11:12 /dev/dsk/c4d0p0
```

התקן שעובדים מולו תו אחרי תו -

```
crw-r----- 1 root sys 262, 16 2010-07-21 11:12 /dev/rdisk/c4d0p0
```

רענון עץ ההתקנים
הפקודה devfsadm מאפשרת רענון של עץ ההתקנים.

devfsadm

devfsadm -v - verbose output

devfsadm -Cv – remove all stale device files.

תרגילי בית

1. מדוע הגרעין לא עובד עם שמות התקן פיזיים?

ניהול דיסקים

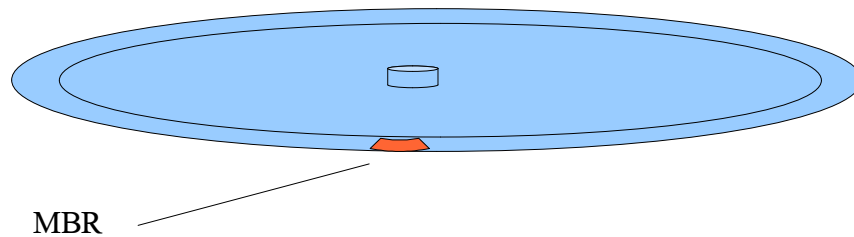
נושאי לימוד:

- טבלת המחיצות – MBR
- טבלת המחיצות - Label/VTOC
- מיקום התקני דיסקים
- הפקודה fdisk
- הפקודה format
- הפקודה rmformat

בפרק זה יילמדו התכנים הבאים:
כיצד בנוי דיסק (סקטורים, מסלולים, צילינדרים). וכיצד לנהל את טבלת המחיצות של סולריס (Vtoc או label) על ידי הפקודה format וכן כיצד לנהל טבלת מחיצות בסביבה מעורבת (Master Boot Reocord) על ידי שימוש בפקודת fdisk. כמו כן, התלמיד ילמד כיצד לנהל טבלת מחיצות על ידי התוכנה gparted, תוכנה חלונאית המאפשרת ניהול מחיצות מתקדם.

MBR – טבלת המחיצות

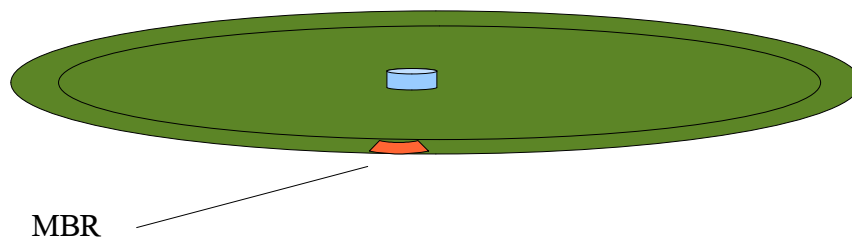
בסביבת ארכיטקטורת X86 ישנה מוסכמה כי בכל דיסק קשיח המחולק למחיצות, סקטור ראשון של הדיסק יכול את טבלת המחיצות הקיימת באותו דיסק קשיח.



מחיצה פיזית – ניתן להגדיר עד 4 מחיצות פיזיות בדיסק קשיח. מחיצה פיזית מכילה מערכת קבצים אחת תחתיה.

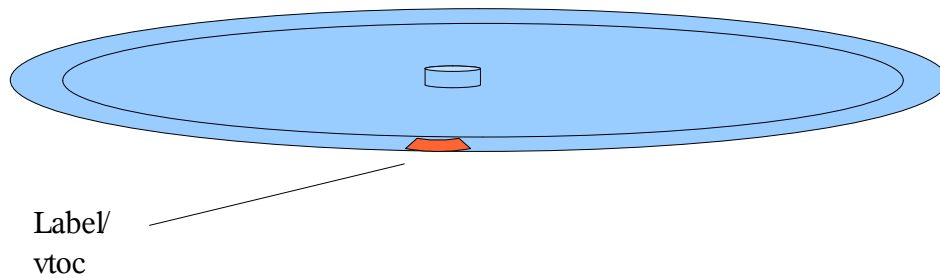
מחיצת הרחבה – על מנת לאפשר הגדרה של יותר מ-4 מחיצות, ניתן להגדיר מחיצת הרחבה (Extended partition) ובתוכה להגדיר מחיצות לוגיות. מחיצת הרחבה אינה מכילה מערכת קבצים תחתיה, אלא מחיצות לוגיות בלבד שבתוכן מערכות קבצים.

מחיצה לוגית – מחיצה הקיימת בתוך מחיצת ההרחבה. ניתן להגדיר כמות בלתי מוגבלת של מחיצות לוגיות, אולם ייתכן כי תהיה מגבלה ברמת מערכת ההפעלה.



LABEL/VTOC – טבלת המחיצות

בסביבת ארכיטקטורת Sparc בדומה לסביבת X86, ישנה מוסכמה כי בכל דיסק קשיח המחולק למחיצות, סקטור ראשון של הדיסק יכיל את טבלת המחיצות הקיימת באותו דיסק קשיח טבלה זו נקראת Label או VTOC (virtual table of content).



טבלת המחיצות Label/Vtoc מאפשרת להגדיר עד 7 מחיצות לשימוש כאשר מחיצה 2 הינה מצביע לכל הדיסק (כלומר מחיצה 2 לא ניתנת לשימוש והיא מוגדרת מחיצה שכוללת את כל הדיסק).
 מחיצה פיזית – ניתן להגדיר עד 4 מחיצות פיזיות בדיסק קשיח. מחיצה פיזית מכילה מערכת קבצים אחת תחתיה.
 מחיצת הרחבה – על מנת לאפשר הגדרה של יותר מ-4 מחיצות, ניתן להגדיר מחיצת הרחבה (Extended partition) ובתוכה להגדיר מחיצות לוגיות. מחיצת הרחבה אינה מכילה מערכת קבצים תחתיה, אלא מחיצות לוגיות בלבד שבתוכן מערכות קבצים.
 מחיצה לוגית – מחיצה הקיימת בתוך מחיצת הרחבה. ניתן להגדיר כמות בלתי מוגבלת של מחיצות לוגיות, אולם ייתכן כי תהיה מגבלה ברמת מערכת ההפעלה.

מיקום התקני דיסקים

סולריס שומר את המידע על דיסקים בשתי ספריות:

ספריה בה נשמרים שמות דיסקים מסוג block devices , אילו דיסקים שבנו עליהם מערכות קבצים ועובדים מולם עם .buffer.

/dev/dsk –

ספריה בה נשמרים שמות דיסקים מסוג char devices , אילו דיסקים שעובדים מולם תו אחרי תו. (Character by character)

/dev/rdisk

מוסכמת שם לדיסקים

הקונבנציה במערכות פתוחות מסוג סולריס הנה ששמות דיסקים יהיו מהצורה – cXtXdXsX כאשר X מייצג מספר כלשהו לפי המוסכמה

c - מספר בקר דיסק (Controller)

t - מספר Target

d - מספר Lun

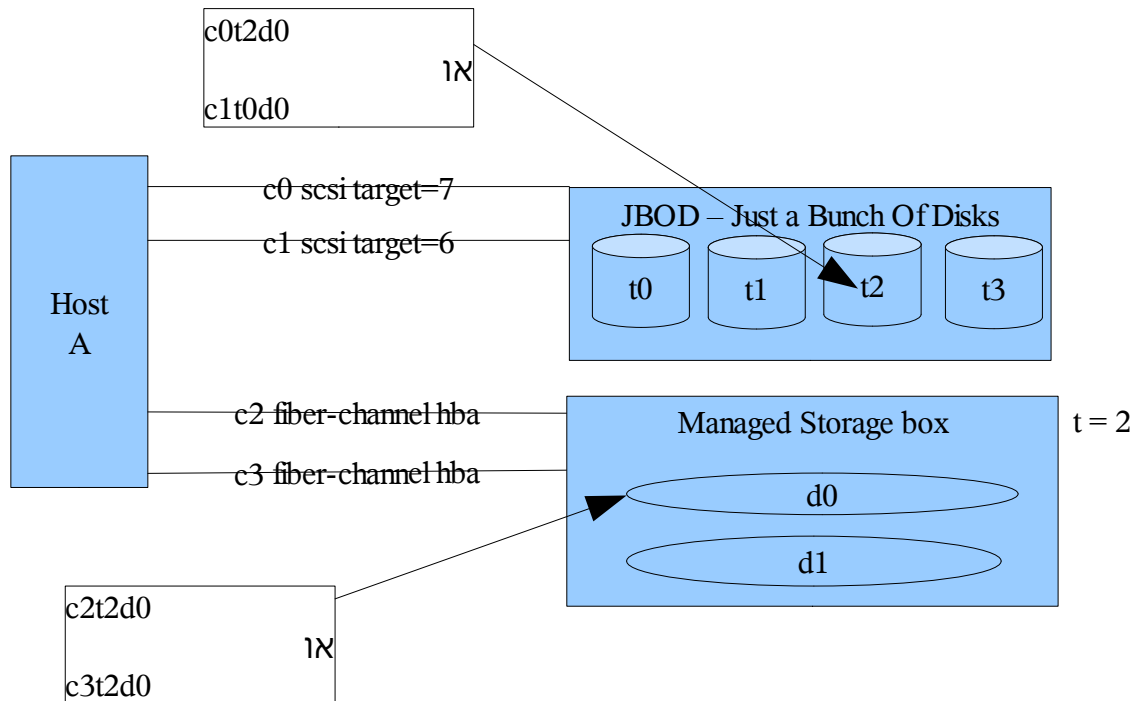
p - מספר Physical partition (מחיצה פיזית ב- MBR)

s - מספר מחיצה לוגית

cXtXdXsX

c0t0d0s0

לדיסק בסולריס יכול להיות יותר משם דיסק אחד מכיוון שיכולים להיות כמה נתיבים לאותו דיסק.



Storage Multipathing

במקרה שיש כמה נתיבים לאותו דיסק ורוצים לעבוד עם שם מאוחד, ניתן להפעיל את הפקודה -

`stmsboot -e`

פקודה זו מוחקת את שני הנתיבים הקיימים ומייצרת שם נתיב אחד המאפשר לעבוד מול אותו שם דיסק לקבלת יתירות (redundancy) ו- Load Balancing

הפקודה fdisk

מאפשרת לנהל את טבלת המחיצות ב-MBR בדומה לכלי הניהול החלונאי .gparted כדי להפעיל את Gparted:

Applications → System Tools → Gparted Partition Editor

הפקודה format

הפקודה format מאפשרת לנהל את הדיסקים שנמצאים בשליטת סולריס דיסקים שמנוהלים תחת סולריס מקבלים .label/vtoc
 Vtoc -Virtual table of content הינו תוכן עניינים וירטואלי שמקבל כל דיסק שמנוהל תחת סולריס.

```
haim@os1:~$ pfexec format
Searching for disks...done
```

AVAILABLE DISK SELECTIONS:

```
0. c4d0 <DEFAULT cyl 2085 alt 2 hd 255 sec 63>
   /pci@0,0/pci-ide@1,1/ide@0/cmdk@0,0
```

```
Specify disk (enter its number): 0
selecting c4d0
```

FORMAT MENU:

```
disk    - select a disk
type    - select (define) a disk type
```

```
.
.
.
```

```
!<cmd>  - execute <cmd>, then return
```

```
quit
```

```
format> p
```

PARTITION MENU:

```
0      - change `0' partition
```

```
.
.
.
```

```
!<cmd> - execute <cmd>, then return
```

```
quit
```

```
partition> p
```

```
Current partition table (original):
```

```
Total disk cylinders available: 2085 + 2 (reserved cylinders)
```

Part	Tag	Flag	Cylinders	Size	Blocks
0	root	wm	3 - 2084	15.95GB	(2082/0/0) 33447330
1	unassigned	wm	0	0	(0/0/0) 0
2	backup	wu	0 - 2084	15.97GB	(2085/0/0) 33495525
3	unassigned	wm	0	0	(0/0/0) 0
4	unassigned	wm	0	0	(0/0/0) 0

5	unassigned	wm	0	0	(0/0/0)	0
6	unassigned	wm	0	0	(0/0/0)	0
7	unassigned	wm	0	0	(0/0/0)	0
8	boot	wu	0 - 0	7.84MB	(1/0/0)	16065
9	unassigned	wm	0	0	(0/0/0)	0

בפועל לא נצטרך להתעסק עם מחיצות לוגיות בסביבת סולריס.

הפקודה **rmformat**

הפקודה **rmformat** מאפשרת לצפות בכל הדיסקים הנתיקים (remote magnetic format) במערכת על מנת למצוא מה מבנה ה- cXtXdXsX שלהם.

התקנת חבילות תוכנה וטלאי מערכת הפעלה

נושאי הלימוד:

- ניהול חבילות
 - חבילות SVR4
 - מנהל החבילות IPS
- ניהול טלאים
- כלי מנהל החבילות
- ניהול סביבות ריצה

בפרק זה יילמדו התכנים הבאים:
מה הרעיון מאחורי שימוש בחבילות תוכנה.
נלמד כיצד להתקין או להסיר חבילות תוכנה משני סוגים
ניהול חבילות מבוסס SVR4 וניהול חבילות מבוסס IPS.
ניהול חבילות מבוסס SVR4 - על ידי שימוש בפקודות pkgadd לשם התקנת חבילת תוכנה ו-
pkgrm לשם הסרת חבילת תוכנה מסוג SVR4.
ניהול חבילות מבוסס IPS - על ידי שימוש בפקודה pkg ומספר תת-פקודות להתקנה הסרה
או צפיה בחבילות.
כמו כן נלמד כיצד להשתמש במנהל החבילות החלונאי על מנת להתקין ולהסיר חבילות.

ניהול חבילות

ישנן 2 סוגי מסגרות עבודה לניהול חבילות בסביבת OpenSolaris.

SVR4 packages

חבילות מסוג System V Release 4 הינן חבילות שנתמכות בסולריס החל מגירסאות ישנות (החל מגרסה 2.5) עד למהדורה האחרונה של OpenSolaris. הפקודות לניהול חבילות מסוג זה הן:

pkginfo - צפיה באילו חבילות מותקנות
 pkgadd – הוספת חבילות
 pkgrm – הסרת חבילות מהמערכת

IPS

מערכת Image Packaging System הנה מערכת ניהול חבילות מתקדמת ומתוכננת להחליף את חבילות התוכנה מסוג SVR4.

הפקודות לניהול חבילות מסוג זה הן:

pkg list [-a] – צפיה ברשימת החבילות המותקנות
 -a - מאפשר צפיה ברשימת כל החבילות כולל אלו שלא מותקנות.
 pkg install – התקנה של חבילה
 pkg uninstall – הסרה של חבילה
 pkg history – מאפשר צפייה בהיסטוריית תחזוקת חבילות

דוגמא להתקנת gdesklets:

```
pfexec pkg install gnome/desklet/gdesklets
```

ניהול טלאים

טלאי תוכנה – Patches, הנה מסגרת עבודה שקיימת בניהול חבילות מסוג SVR4. ב-OpenSolaris אין נושא של התקנת טלאים.

הפקודות לניהול טלאי חבילות הן:

showrev -p – צפיה ברשימת הטלאים המותקנים
 patchadd <xxxxxx-yy> – הוספת טלאי תוכנה
 patchrm <xxxxxx-yy> – הסרת טלאי תוכנה

כלי מנהל חבילות

על מנת להפעיל את כלי מנהל החבילות:

System → Administration → Package Manager

מנהל החבילות מאפשר הורדה והתקנה אוטומטית של חבילות תוכנה תוך כדי הורדת והתקנת תלויות (חבילות התוכנה בהן תלויות אותן חבילות שבחרנו).

נעילת מסד נתוני החבילות

לא ניתן לנהל חבילות בו זמנית דרך 2 כלי ניהול שהם (בין אם הם חלונאיים או ממשק פקודה) ברגע שהתחלנו בפעילות מול מסד נתוני החבילות, כל כלי ניהול נוסף שינסה לגשת למסד הנתונים - לניהול החבילות, יודיע כי מסד הנתונים תפוס וינסה להמתין לשחרור שלו.

ניהול סביבות ריצה

סביבת ריצה מאפשרת לשמור את תצורת מערכת ההפעלה ותצורת מערכת הקבצים בזמן נתון. באופן מעשי ניתן להריץ את הפקודה בכל זמן נתון ולחזור לסביבת ריצה במקרה של תקלה או בעיות עם סביבת ריצה הנוכחית. (לדוגמא בגלל וירוס).

```
beadm list  
pfexec beadm create opensolaris-1
```

על מנת להפעיל סביבת ריצה ניתן להשתמש בפקודה -

```
beadm activate opensolaris-1
```

על מנת להכניס שינויים לתוקף יש לבצע אתחול למחשב דרך הפקודה -

```
init 6
```

תרגילי בית

1. בנה סקריפט שמתקין על 10 מחשבים ברשת שכתובותיהם 192.168.1.110-119 את החבילה `gdesklets`.

```
#!/bin/bash
```

```
for I in 110 111 112 113 114 115 116 117 118 119
do
ssh 192.168.1.$I pfexec pkg install gnome/desklet/gdesklets
done
```

ניהול מערכת הקבצים ZFS

נושאי לימוד:

- הסבר על מבנה ZFS
- ניהול מאגרים
- ניהול מערכות קבצים
- ביצוע שינוי סיסמא – ע"י אתחול מ-Live CD
- עבודה עם קבצים כדיסקים

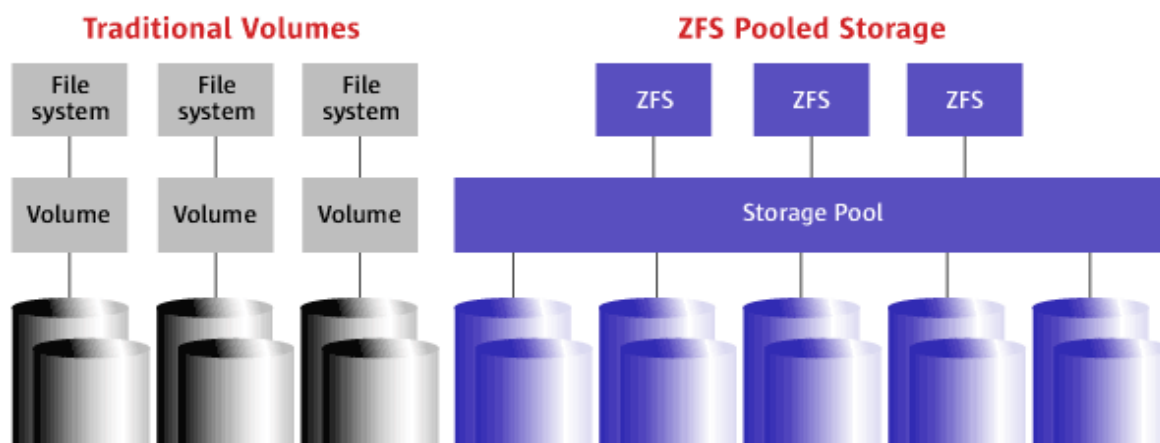
בפרק זה יילמדו התכנים הבאים:

מבנה מערכת הקבצים באופן כללי. כיצד לנהל מאגרים (מאגרי דיסקים - pools) – כיצד לבנות מאגרי דיסקים המספקים יתירות ושרידות מסוג RAID1 בו כל שני דיסקים (או יותר) מהווים העתק זהה אחד של השני או מסוג RAID5/6 בו ישנו ניתן לבנות מערך של כמה דיסקים הסובל כשל של דיסק אחד או שני דיסקים. כמו כן התלמיד ילמד כיצד לנהל מערכות קבצים ונפחי אחסון (data sets) ולהוסיף להן תכונות כגון דחיסה אוטומטית של כל מערכת הקבצים.

כיצד ליצור תמונות של מערכת הקבצים (snapshots) לצרכי גיבוי ואחזור מידע וכיצד לשבט מערכות קבצים (clones) על מנת לאפשר עבודה משותפת על מערכת קבצים כך שלכל משתמש יש העתק פרטי משלו של מערכת הקבצים.

הערה: ניהול מערכת הקבצים שיילמד יכלול את לימוד ממשק הפקודה לניהול ZFS, בעזרת הפקודה zpool – לניהול מאגרים והפקודה zfs לניהול מערכות קבצים.

הסבר על מבנה ZFS



ZFS הינה דרך מהפכנית לניהול מערכות קבצים. עם תכונות ויתרונות שלא נמצאים כרגע בשום מערכת קבצים וכלי לניהול נפחי אחסון.

ZFS מתייחס לניהול אחסון באותה צורה בה מתייחס גרעין מערכת ההפעלה למשאב הזכרון RAM.

כפי שבזכרון RAM, משתמשים ואפילו מנהל המערכת אינם יודעים מהו מבנה הזכרון, כאשר הם עובדים עם זכרון, כך גם ב-ZFS. ZFS מסלק את הצורך בניהול נפחי אחסון. הדיסקים הקשיחים שנמצאים במערכת ההפעלה מוגדרים פעם אחת ומרגע זה, הם מוכנים לשימוש.

אין צורך להגדיר נפח אחסון מסוג RAID1/0/5/6 ולאחר מכן לבנות עליו מערכת קבצים. ZFS דואג לכל באופן אוטומטי. ZFS מאפשר לקבץ (aggregate) כמה מבנים מסוג RAID כלשהו למאגר דיסקים אחד גדול המוכן לשימוש מרגע היצירה שלו ובכך מאפשר פשוט וחוסך מורכבות.

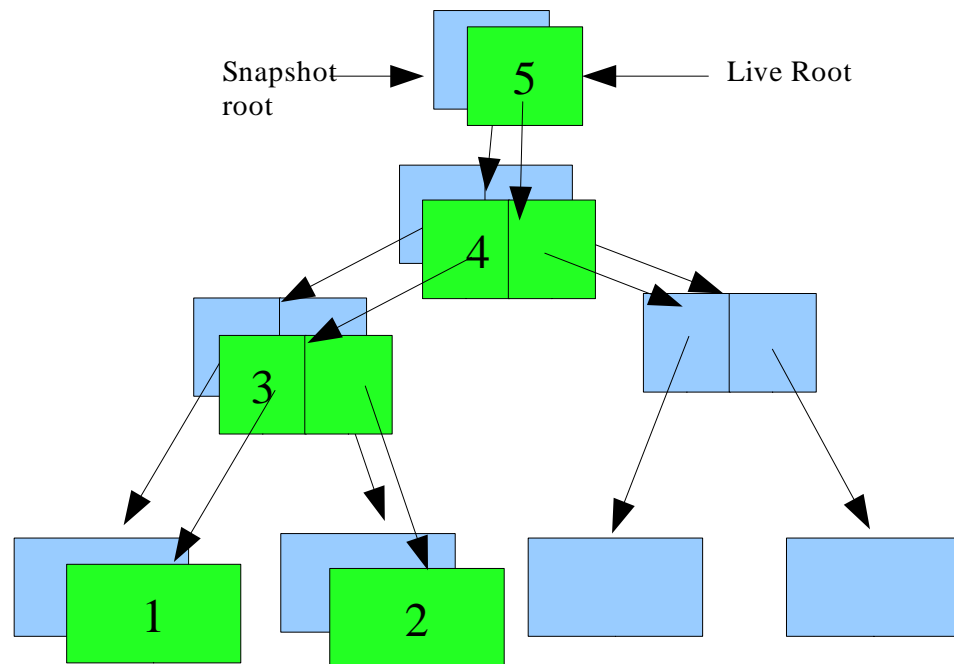
ZFS של תכונות

- מערכת קבצים טרנזקציונלית
 - שימוש במנגנון Copy-on-write
 - לא צריך file-system check-
- התאוששות עצמית
 - בדיקת Checksum
 - שלמות מקצה לקצה
 - התאוששות עצמית למידע יתיר
- תמיכה ב- 128 ביט
- ניהול פשוט
- תמיכה במירב סוגי השיתוף והפרוטוקולים הקיימים -
NFS, CIFS (Solaris express/OpenSolaris), iSCSI, Lun provisioning
(Solaris Express/OpenSolaris via COMSTAR)
- כמות נכבדת של תכונות

מערכת קבצים טרנזקציונית

ZFS הנה מערכת קבצים בה כל אובייקט נכתב בצורת טרנזקציה, כלומר, המידע על הדיסק תמיד נשאר עקבי ללא פגמים, כאשר הוא בדיסק.

מידע לעולם לא נכתב לאזור בדיסק קשיח שכבר מכיל מידע פעיל. רצף של פעולות יבוצע במלואו, או שיתעלמו ממנו אם הוא לא מלא. זה מאפשר לוותר על הצורך בבדיקות מבחץ על מערכת הקבצים, מכיוון שמערכת הקבצים תמיד עקבית בדיסק קשיח.

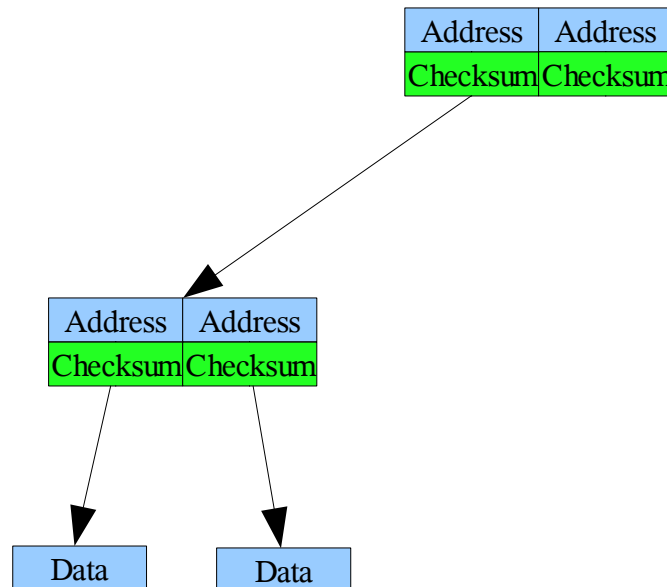


סדר הפעולות הנו מ-1 עד 5. כאשר רק כאשר מתבצעת פעולה 5 כל יתר הפעולות יהיו בתוקף.

למעשה, אם נפעיל Snapshot על מערכת הקבצים הנוכחית – אנו חוסכים את פעולות המחיקה ועדכון מצביעים ובלוקים המתלוות לטרנזקציה של עדכון ה-Uberblock.

התאוששות עצמית

ב-ZFS כל המידע והמטה מידע עובר בדיקות checksum על ידי אלגוריתם שבוחר המשתמש (אלגוריתם ברירת מחדל הוא Fletcher2 שהינו קל יחסית) כל בדיקות ה-checksum וההתאוששות נעשים בשכבת מערכת הקבצים, והדבר שקוף למשתמש.

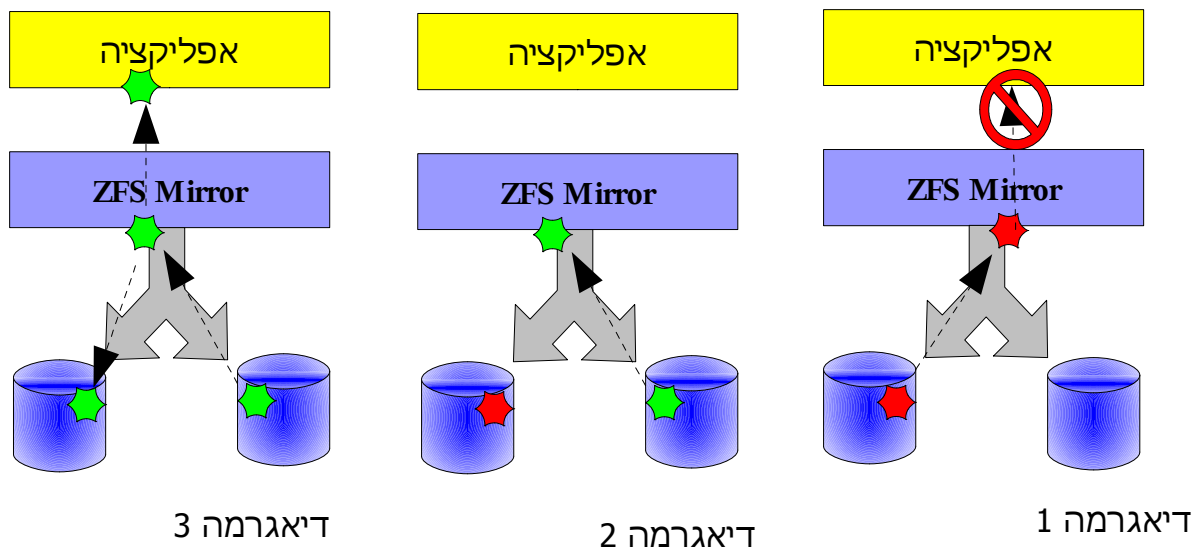


ZFS מאפשר בדיקה של מסלול קלט/פלט מלא ומאפשר מניעת הבעיות הבאות:

- ◆ Bit rot - ביט רקוב (בדרך כלל במדיה או בשבבים)
- ◆ Phantom writes - בעיות בראש הדיסק
- ◆ Misdirected reads and writes - קריאות או כתיבות מוטעות
- ◆ DMA parity errors - שגיאות חישוב DMA
- ◆ Driver bugs - באגים בדרייברים
- ◆ Accidental overwrites - טעויות בכתיבה

התאוששות עצמית למידע יתיר

בנוסף לבדיקת checksum, יש מנגנון של התאוששות עצמית למידע. כאשר יש בלוק לא תקין – ZFS ינסה לחפש מידע תקין באחד ההעתקים הנוספים שקיימים וינסה לתקן את ההעתק הפגום, ולהחליף אותו בהעתק תקין.



אפליקציה שולחת בקשה לקריאת בלוק מידע.

דיאגרמה 1: המידע נבדק על ידי בדיקת Checksum ומתגלה כשגוי.

דיאגרמה 2: ZFS מנסה למצוא העתק של אותו בלוק מידע ולהביא את הבלוק מהעתק תקין.

דיאגרמה 3: ZFS מעביר את ההעתק התקין לאפליקציה ובנוסף מתקן את ההעתק שהתגלה כשגוי.

ניהול מאגרים

ZFS מאפשר ניהול מאגרים כאשר היחידות בתוך המאגרים הינם מחיצות או דיסקים מלאים, כאשר שימוש בדיסק מלא כמובן מומלץ יותר.

גודל מינימלי של יחידה בתוך מאגר צריך להיות גדול מ-64MB.

כאשר עובדים עם דיסקים מלאים, ZFS מפרמט את הדיסק עם טבלת מחיצות מסוג EFI. (זוהי טבלת מחיצות שמתאימה ל-BIOS מסוג EFI). ומשתמש במחיצה 0 בדרך כלל כשמחיצה זו מכילה את כל תוכן הדיסק.

כל מאגר שנוצר על ידי ZFS מקבל מספר ייחודי משל עצמו הנקרא GUID. כל דיסק במאגר מקבל גם GUID ייחודי משל עצמו, כך ש-ZFS אינו משתמש במיקום הדיסק על מנת למצוא אילו דיסקים מרכיבים את המאגר.

Virtual Device – Vdev

ZFS מאפשר יצירת Vdev מהסוגים הבאים:

no vdev – Striping

mirror – RAID1 - Mirror

raidz – Like RAID5 with single bit parity.

raidz2 – Like RAID6 with dual bit parity.

כל מבנה שאינו מכיל יתירות יהיה במבנה של Stripe. ZFS ינסה ליצור עם הזמן מבנה של נתונים שיאפשר Striping בין Vdev's במאגר.

הרצת סימולציה

`zpool <sub-command> -n` – will simulate dry run operation to simulate the operation on the # pool without actually writing data to disk

יצירת מאגר מסוג Stripe

על ידי שימוש בפקודה `zpool create`.

```
# zpool create [-f] my-stripe disk1 disk2 disk3
```

This command will generate a warning if any disk is already use by the system. You may override

any warning notifications by using the `-f` option.

הוספת דיסק נוסף למאגר מסוג Stripe

```
# zpool add my-stripe disk4
```

After adding disk4 to the pool, ZFS will try to balance the data between all stripe members by using the copy-on-write mechanism to allow a new striping layout consisting of disk4.

יצירת מאגר מסוג mirror

```
#zpool create my-mirror mirror disk1 disk2 mirror disk3 disk4 spare disk7
```

הוספת Mirror נוסף למאגר מסוג mirror

```
#zpool add my-mirror mirror disk5 disk6
```

אחרי הוספת מבנה מסוג mirror למאגר, ZFS ינסה לבצע איזון בין המידע הנמצא בין כל המבנים במאגר על ידי מנגנון Copy-on-write.

המרת מאגר מסוג Stripe למאגר מסוג Mirror

ההמרה נעשית על ידי שימוש בפקודה `zpool attach`.

כאשר disk1 הנו דיסק שנמצא במאגר ואליו מוסיפים את disk4 על מנת ליצור מבנה מסוג mirror. וכך גם לגבי דיסקים disk2 ו-disk3.

```
#zpool attach my-stripe disk1 disk4
```

```
#zpool attach my-stripe disk2 disk5
```

```
#zpool attach my-stripe disk3 disk6
```

המרת מאגר מסוג mirror למאגר רגיל ללא יתירות

על ידי שימוש בפקודה `zpool detach` המאפשרת ניתוק של מבנה mirror.

```
#zpool dettach my-stripe disk1
```

```
#zpool dettach my-stripe disk2
#zpool dettach my-stripe disk3
```

יצירת מאגר מסוג RAIDZ1 (RAID5)

```
#zpool create my-raidz raidz disk1 disk2 disk3 raidz disk4 disk5 disk6 spare
\ disk10
```

הוספת מבנה raidz למאגר קיים

```
#zpool add my-raidz raidz disk7 disk8 disk9
```

After adding the new raidz virtual device (consisting of disk7, disk8 and disk9), ZFS will balance .data between all virtual devices to allow a new striping layout consisting of the new virtual device

יצירת מאגר מסוג RAIDZ2

```
#zpool create my-raidz2 raidz2 disk1 disk2 disk3 disk4 raidz2 disk5 disk6 \
disk7 disk8 spare disk13
```

הוספת מבנה raidz2 נוסף למאגר הקיים -

```
#zpool add my-raidz2 raidz2 disk9 disk10 disk11 disk12
```

After adding the new raidz2 virtual device (consisting of disk9, disk10, disk11 and disk12), ZFS will balance data between all virtual devices to allow a new striping layout consisting of the new virtual device.

הריסת מאגר

ניתן להרוס מאגר על ידי הפקודה `zpool destroy`.

שימו לב, הפקודה הינה הרסנית במיוחד ויכולה למחוק מאגרים גם אם הם בשימוש.

```
#zpool destroy [-f] my-raidz2
```

You may destroy storage pools with faulty devices by using the `-f` option.

כיצד לצפות בסטטוס של מאגר

ניתן לצפות בסטטוס של מאגר ולראות מאיזה דיסקים הוא מורכב על יש הפקודה -

```
zpool status
```

צפיה במידע על המאגר

על ידי שימוש בפקודה `zpool list`

ללא ארגומנטים הפקודה מציגה סטטוס של כל המאגרים

```
#zpool list
```

ניתן לקבל סטטוס על מאגר בודד על ידי רישומו כארגומנט בפקודה

```
#zpool list my-stripe
```

ניתן לבצע סקריפטים על ידי שימוש באופציה `-o`.

```
#zpool list -o name,size my-stripe
```

ניתן להוריד את שורת ה-Header על ידי האופציה `-H`.

צפיה בסטטיסטיקות על המאגר

צפיה בסטטיסטיקות על המאגר נעשית על ידי הפקודה `zpool iostat`

ללא ארגומנטים הפקודה מציגה סטטיסטיקות על כל המאגרים.

ניתן להוסיף את האופציה `-v` אם רוצים לראות מידע גם על Virtual devices.

```
#zpool iostat [-v]
```

ניתן לקבל סטטיסטיקות על מאגר בודד על ידי רישומו כארגומנט בפקודה

```
#zpool iostat my-stripe
```

ניתן לקבל מידע בזמן אמת על ידי רישום אינטרבל זמן (ארגומנט שני) וכמות פעמים (ארגומנט שלישי).

```
#zpool iostat my-stripe 2 3
```

בדיקת תקינות מאגר

בדיקת תקינות מאגר נעשית על ידי הפקודה `zpool status -x`.

לשם קבלת יותר מידע ניתן להוסיף את האופציה -v.

```
#zpool status -v my-stripe
```

כיצד לייבא וליצא מאגרים

במקרה שעובדים עם רשת אחסון אזורית (SAN – Storage area network) ניתן לבצע ייבוא או ייצוא של מאגרים. לכל מאגר יש מזהה יחודי – GUID (Global Unique ID)

הסתגלות לסדר סיביות (**Adaptive endianness**)

מיגרציה של מאגר בין ארכיטקטורות שונות אפשרית. ZFS הינו נייטרלי לסדר סיביות בין ארכיטקטורות RISC לארכיטקטורות CISC וידע לתמוך בשתי הארכיטקטורות.

ארכיטקטורת RISC – Little endian

ארכיטקטורת CISC – Big endian

ייצוא של מאגר (התנתקות ממאגר)

על מנת לייצא מאגר ניתן להשתמש בפקודה `zpool export`. ניתן להכריח ייצוא מאגר על ידי שימוש באופציה `-f`.

```
#zpool export [-f] my-stripe
```

ייבוא של מאגר (התחברות למאגר)

ניתן לייבא מאגר על ידי הפקודה `zpool import`.

על מנת לדעת איזה מאגרים ניתנים לייבוא נשתמש בפקודה ללא ארגומנטים -

```
#zpool import
```

על מנת לייבא מאגר מסויים השתמש בפקודה -

```
#zpool import my-stripe
```

על מנת לשנות שם למאגר השתמש בפקודה -

```
#zpool import <old-name|GUID> <new-name>
```

ניתן להתאושש מהריסה של מאגר

ניתן להשתמש בפקודה `zpool import -D` על מנת לגלות מאגרים שנהרסו.

```
#zpool import -D
```

על מנת לייבא מאגר שנהרס השתמש בפקודה -

```
#zpool import -Df my-stripe
```

ניהול מערכות קבצים

כיצד ליצור ולהרוס מערכות קבצים

כאשר נוצר מאגר נוצרת גם מערכת קבצים עליונה בראש המאגר והיא מעוגנת לראש מערכת הקבצים הראשית במערכת ההפעלה. תחת ספריית השורש - /, לדוגמא אם שם המאגר הוא rpool המאגר יעגן בספרייה ששמה - /rpool.

יצירת מערכת קבצים

יצירת מערכת קבצים נעשית על ידי הפקודה `.zfs create`. ניתן להמשיך ליצור מערכות קבצים ברמה העליונה של המאגר. לדוגמא יצירת מערכת הקבצים - `my-stripe/home`

```
#zfs create my-stripe/home
```

ZFS אוטומטית עוגן את מערכת הקבצים שנוצרה. ספריית העגינה תיקרא בדרך כלל בשם נתיב של מערכת הקבצים שנוצרה תחת המאגר עם הוספת / על מנת להגיע למערכת הקבצים מספריית השורש - /.

ניתן ליצור שרשרת של מערכות קבצים על ידי שימוש באופציה המוכרת `-p`.

לדוגמא:

```
#zfs create -p my-stripe/fs1/fs2/fs3
```

שינוי שם למערכת קבצים

שינוי שם למערכת הקבצים נעשה על ידי הפקודה - `.zfs rename`

ניתן לשנות שם של מערכת קבצים לדוגמא:

```
#zfs rename my-stripe/home my-stripe/old-home
```

ZFS ינסה לנתק את מערכת הקבצים עם השם הישן ולעגון את מערכת הקבצים עם השם החדש תחת המיקום בהיררכיה המאגר החדשה.

הריסת מערכות קבצים

ניתן להרוס מערכת קבצים על ידי הפקודה `.zfs destroy` על מנת להרוס מערכת קבצים:

```
#zfs destroy [-f] my-stripe/home
```

ניתן להרוס מערכת קבצים בצורה ברוטלית על ידי שימוש באופציה `-f`.

האופציה `-r` מאפשרת להרוס מערכת קבצים וכל הצאצאים שלה.

האופציה `-R` מאפשרת להרוס מערכת קבצים

`R` option will allow destroying of a file-system with all of its indirect dependents. (Use with extreme caution)

קבלת מידע על מערכת קבצים

The `zfs list` command allows querying of file-system information

`.r` option will allow querying a file-system within specific pool or hierarchy-

לדוגמא:

```
# zfs list -r my-stripe
```

כיצד לצפות ולשנות תכונות של מערכת קבצים

ניתן לצפות בתכונות של מערכת הקבצים על ידי הפקודה `.zfs get`

```
#zfs get all my-stripe/home
```

ניתן לצפות גם בתכונה מסוימת של מערכת קבצים לדוגמא צפיה בתכונה `mountpoint`:

```
#zfs get mountpoint my-stripe/home
```

ישנם תכונות שניתנות לקריאה בלבד כגון: `available, creation, mounted, origion, compressratio, referenced, type, used`

ישנן תכונות שניתנות לקריאה וכתיבה כגון: aclinherit, aclmode, atime, checksum, compression, devices, exec, mountpoint, quota, readonly, recordsize, reservation, sharenfs, sharesmb, setuid, .snapdir, volsize, volblocksize, zoned

רוב התכונות שניתנות לקריאה/כתיבה נורשות ממערכת הקבצים העליונות במאגר.

ניתן לשנות תכונה במערכת הקבצים על ידי הפקודה `zfs set`.

לדוגמא הפעלת התכונה קריאה בלבד למערכת הקבצים

```
#zfs set readonly=on my-stripe/home/software
```

לדוגמא הפעלת שיתוף מערכת הקבצים על ידי פרוטוקול NFS.

```
#zfs set sharenfs=on my-stripe/home/haim
```

לדוגמא הפעלת שיתוף מערכת הקבצים על ידי פרוטוקול CIFS (שיתוף עם מערכות Windows).

```
#zfs set sharesmb=on my-stripe/home/haim
```

מכסות ושמירת מקום - quotas and reservations

מכסה – Quata

שמירת מקום – Reservation

יצירת מכסה נעשית על ידי הפקודה -

```
zfs set quota=
```

לדוגמא:

```
#zfs set quota=50G my-stripe/home/haim
```

יצירת שמירת מקום נעשית על ידי הפקודה -

```
zfs set reserv=
```

לדוגמא:

```
#zfs set reserv=1G my-stripe/home/haim
```

Reservation -1 Quota הם מספרים לוגיים בלבד וניתן ליצור מכסות או שמירות מקום עם מספרים הגדולים מגודל המאגר על מנת לאפשר אחיפה עתידית.

copies

copies מאפשר יצירת מספר עותקים למערכת קבצים (ללא תלות במבנה המאגר)

```
#zfs set cpoies=2 my-stripe/home/haim
```

ניתן ליצור עד 3 העתקים למערכת קבצים.

כיצד ליצור ולהרוס snapshots

יצירת snapshot

יצירת Snapshot נעשית על ידי הפקודה `.zfs snapshot`

שם של Snapshot נראה כך:

`filesystem@snapname`

`volume@snapname`

יצירת Snapshot לדוגמא:

```
#zfs snapshot my-stripe/home/haim@friday-10-05-2009
```

הריסת snapshot

ניתן להרוס Snapshot על ידי הפקודה `.zfs destroy`

לדוגמא:

```
#zfs destroy my-stripe/home/haim@friday-10-05-2009
```

גלגול ל – snapshot קודם

כדי לראות אילו snapshots קיימים ניתן להשתמש בפקודה -

```
#zfs list -t snapshot
```

כדי לשחזר מערכת קבצים למידע שהיה קיים ב-Snapshot מסוים -

```
#pfexec zfs rollback [-r] my-stripe/home/haim@friday-10-05-2009
```

שים לב, כי גלגול ל-Snapshot קודם ימחק למעשה את כל ה-Snapshot-ים החדשים.

יצירה והריסת שיבוטים clones

ניתן לשבט Snapshot להיות מערכת קבצים מלאה על ידי הפקודה `.zfs clone`.
לדוגמא:

```
#zfs snapshot my-stripe/home/haim@friday-10-05-2009
```

```
#zfs clone my-stripe/home/haim@friday-10-05-2009 my-stripe/haim-friday-10-05-2009
```

הריסה של Clone נעשית על ידי הפקודה `.zfs destroy`.

```
#zfs destroy my-stripe/haim-friday-10-05-2009
```

קידום של- clone

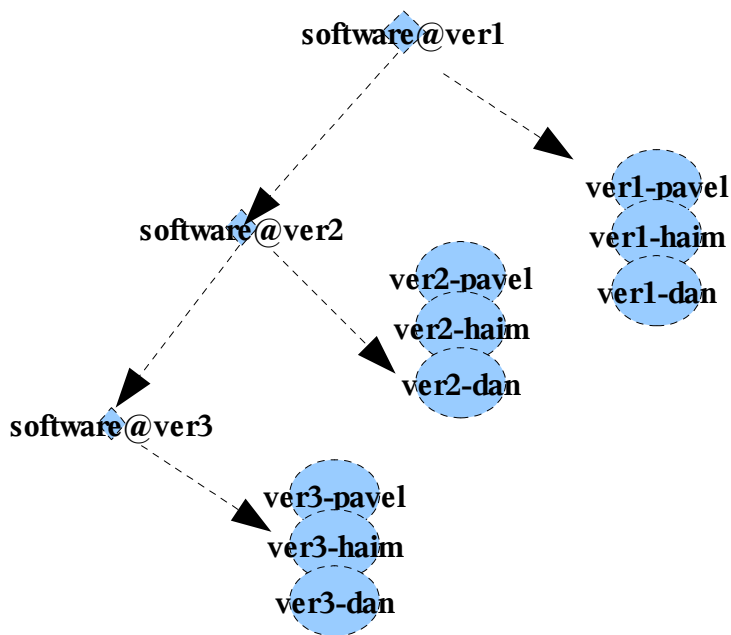
ניתן להעלות Clone דרגה כך שיהיה בלתי תלוי ב-Snapshot המקורי ממנו הוא נוצר. זה מאפשר גם להרוס את מערכת הקבצים ממנו נוצר אותו Clone. וכך לשנות שם ל-Clone כך שייקרא על שם מערכת הקבצים ממנו נוצר.

דוגמא לקידום:

```
#zfs promote my-stripe/haim-friday-10-05-2009
```

דוגמא למערכת קבצים משובטת:

- .ver1 – Snapshot בשם
 - שדרוג לגרסה שניה שלה ויצירת Snapshot נוסף – .ver2
 - שדרוג לגרסה שלישית שלה ויצירת Snapshot נוסף – .ver3
- כעת ניתן לשבט כל אחד מה-Snapshots למשתמשים השונים ללא פגיעה ב-Snapshots עצמם או במערכת הקבצים שממשיכה להתקדם.



עבודה עם קבצים כדיסקים

בנה נפח אחסון (Volume) על ידי הפקודה -

```
pfexec zfs create -V 2g rpool/vol1
```

בנה מערכת קבצים UFS על הנפח אחסון rpool/vol1 על ידי הפקודה -

```
pfexec newfs /dev/zvol/rdisk/rpool/vol1
```

הקש y על מנת ליצור את מערכת הקבצים.

עגון את מערכת הקבצים להיררכיית הקבצים תחת ספריית השורש על ידי הפקודה -

```
pfexec mkdir /mnt/vol1
```

```
pfexec mount rpool/vol1 /mnt/vol1
```

יצירת קבצי דיסקים על ידי הפקודה -

```
pfexec mkfile 100m /mnt/vol1/disk1 /mnt/vol1/disk2 /mnt/vol1/disk3
```

כעת ניתן ליצור מאגרים על ידי אספקה של שם הדיסק המלא לדוגמא:

```
zpool create mypool /mnt/vol1/disk1 /mnt/vol1/disk2
```

רצף האתחול

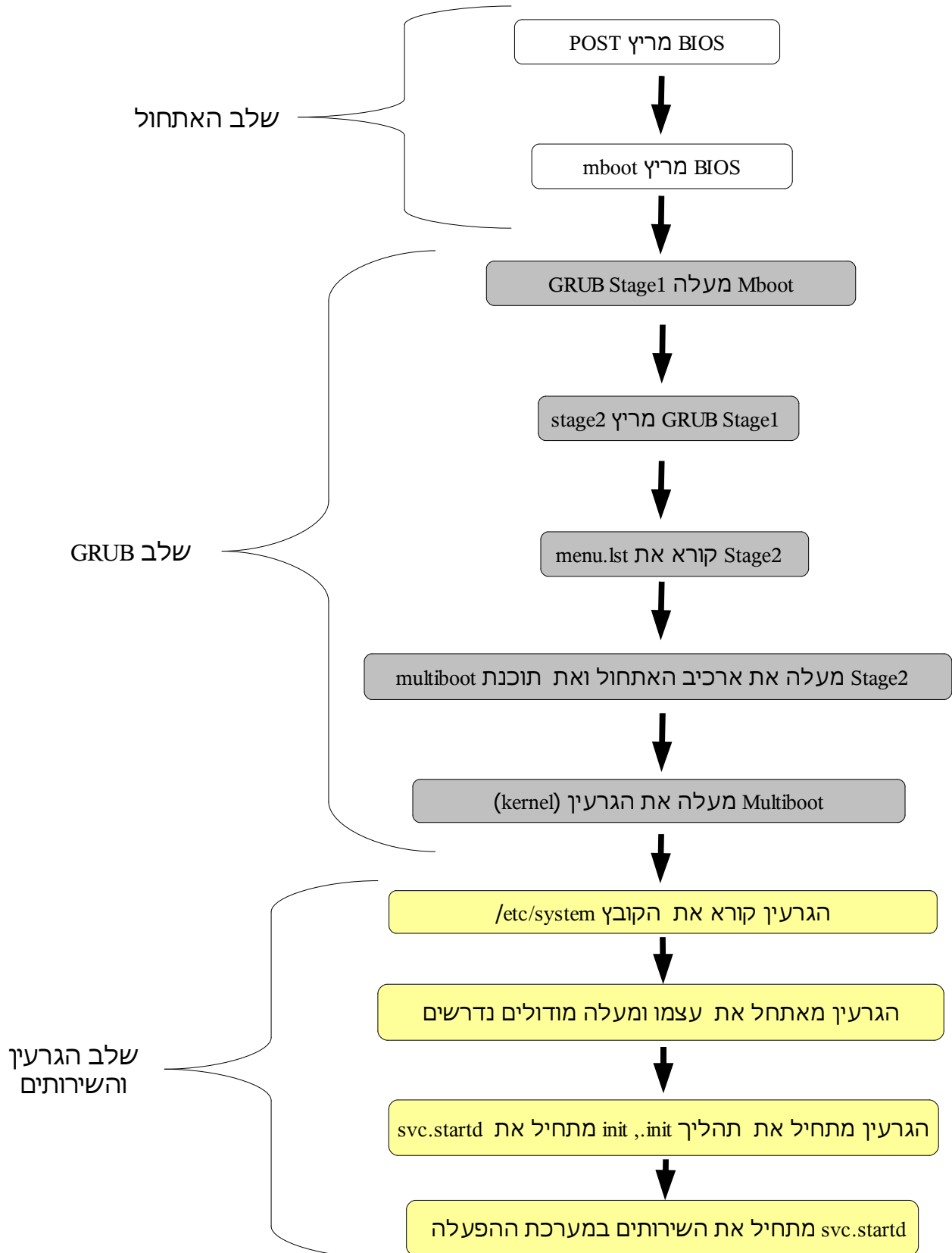
נושאי לימוד:

- תהליך האתחול במחשב
- מהו GRUB
- אפשרויות מתקדמות
- הפקודה bootadm
- קביעת מערכת הפעלה ברירת מחדל
- קביעת השהייה ב- GRUB
- התאוששות מהגדרות משובשות

בפרק זה יילמדו התכנים הבאים:
מהו רצף תהליך האתחול לאחר האתחול על ידי GRUB.

כיצד תהליך מתחיל תהליך האתחול בסולריס החל משלב הדלקת המחשב ביצוע רוטינת Power On Self Test ומעבר להרצת המאתחל Grub המאפשרת אתחול של מספר מערכות הפעלה המותקנות יחד על אותו מחשב (כגון חלונות, סולריס ולינוקס).
כיצד לשלוט בתכונות של GRUB על ידי הפקודה bootadm וכיצד לאפשר בחירת אתחול אוטומטי של אחת מהמערכות הפעלה המותקנות - עם השהייה (Timeout) קצרה.
נלמד כיצד לאפשר עליה בשלבים למקרה בו מערכת ההפעלה נתקעת באמצע עלייתה.
(על ידי עריכת אופציות האתחול והוספת האופציה -a בפקודת kernel, לתהליך האתחול).
וכן כיצד לשנות הגדרות בסולריס במקרה ומערכת הקבצים מכילה הגדרות משובשות (כגון במקרה וסיסמת root אינה ידועה, דבר המונע התחברות ועבודה על המערכת).

תהליך האתחול במחשב – POST



שלב האתחול

ה- POST הינה שגרה הנמצאת ב- BIOS של המחשב. POST בודק את מצב החומרה – מעבדים, זכרון וכו'. אם לא נמצאת שגיאה POST מסיים ומריץ את השגרה mboot.

במקרה ונמצאו שגיאות שאינן קריטיות, הוא ידווח הודעה על המסך וימשיך להריץ את mboot. במקרה והשגיאות קריטיות לתפעול המערכת, POST עוצר ומדווח שגיאה מבלי להמשיך.

שלב ה- GRUB

שגרת mboot מעלה את Grub stage1 שנמצאת בדרך כלל ב- MBR. Grub stage1 מעלה את Grub stage2. בדרך כלל stage2 הינו שלב בו יש תוכנה המודעת למבנה מערכת הקבצים, תוכנה זו יודעת היכן נמצא קובץ ההגדרות menu.lst של תוכנת ה- Grub והיא קוראת אותו.

Grub stage2 קוראת איזה שורה מתוך קובץ ההגדרות menu.lst לעלות ומעלה את הגרעין של אותה מערכת הפעלה על ידי הפעלת תוכנת multiboot. תוכנת multiboot אחראית להריץ את הגרעין.

שלב הגרעין והשירותים

הגרעין קורא את קובץ ההגדרות /etc/system ומאתחל את עצמו ומעלה מודולים נדרשים. הגרעין נעזר בתוכנת zfsboot על מנת להיות מודע איפה קבצי המודולים. הגרעין מאתחל את התהליך /etc/init. התהליך init מאתחל את svc.startd. svc.startd מאתחל את שירותי מערכת הפעלה.

מהו GRUB

Grand Unified Boot Loader – Grub הנה תוכנה המאפשרת העלאה של מספר מערכות הפעלה המותקנות על גבי מדיה במערכת.

ל-Grub יש שני שלבי הפעלה stage1 ו-stage2 המאפשרים את עליית grub וטעינת קובץ ההגדרות menu.lst. ניתן לעצור את grub בשלב העליה ולבחור אחת מהאופציות שבתפריט על מנת להעלות מערכת הפעלה אחרת מהמערכת הפעלה שמוגדרת כברירת מחדל.

שלב stage1

stage1 הנו שלב המותקן בדרך כלל ב-MBR או במחיצה האקטיבית שמוגדרת ב-MBR.

שלב stage2

stage2 הנו שלב שמותקן בדרך כלל בתוך המחיצה האקטיבית והנה תוכנה שיודעת לקרוא את מבנה מערכת הקבצים שמוגדרת במחיצה האקטיבית.

קובץ menu.lst

קובץ menu.lst הנו קובץ הגדרות טקסטואלי של Grub. בקובץ זה נקבעת איזה מערכת הפעלה היא הברירת מחדל – על ידי המשתנה default. מה הזמן timeout עד לעליה אוטומטית של מערכת הפעלה ברירת המחדל – על ידי המשתנה timeout, והגדרות נוספות.

אפשרויות מתקדמות

על מנת לערוך את האפשרויות שב-grub. מעלים את grub ולוחצים מיד מקשי חץ למעלה או מטה. בוחרים את השורה הרצויה ולוחצים e על מנת לערוך את הפקודות שבאותה שורה. לאחר סיום עריכת אותה שורה לוחצים Enter על מנת לשמור שינויים ולוחצים b כדי להריץ את הבחירה הנוכחית עם השינויים שביצעתי.

עליה אינטראקטיבית

דוגמא אם רוצים עליה אינטראקטיבית (-a):
 בוחרים את השורה הרצויה ולוחצים e על מנת לערוך את הפקודות שבאותה שורה.
 מוחקים את השורות המתחילות במילים: splashimage, foreground, background על ידי לחיצה על d.
 בוחרים את שורת kernel ולוחצים e כדי לערוך אותה. משנים console=text ומוסיפים את האופציה -a בסוף השורה. לאחר סיום עריכת אותה שורה לוחצים Enter על מנת לשמור שינויים ולוחצים על b כדי להריץ את הבחירה הנוכחית עם השינויים שביצעתי.

עליה למצב תחזוקה

אם רוצים לעלות למצב תחזוקה – Single user mode
 בוחרים את השורה הרצויה ולוחצים e על מנת לערוך את הפקודות שבאותה שורה.
 מוחקים את השורות המתחילות במילים: splashimage, foreground, background על ידי לחיצה על d.
 בוחרים את שורת kernel ולוחצים e כדי לערוך אותה. משנים console=text ומוסיפים את האופציה -s בסוף השורה. לאחר סיום עריכת אותה שורה לוחצים Enter על מנת לשמור שינויים ולוחצים על b כדי להריץ את הבחירה הנוכחית עם השינויים שביצעתי.

חשוב ! - אין להשאיר שורות ריקות בשורת הפקודה של grub .

הפקודה bootadm

הפקודה bootadm מאפשרת ניהול של grub ושליטה על התכונות השונות שלו.
 הפקודה מודיעה היכן נמצא הקובץ הגדרות menu.lst הפעיל במערכת.

במקרה שלנו -
 /rpool/boot/grub/menu.lst

כדי לצפות ברשימה כפי שמופיעה ב-grub יש להשתמש בתת הפקודה list-menu:
 bootadm list-menu [-R altroot]

לדוגמא:

```
haim@opensolaris:/boot/grub$ bootadm list-menu
the location for the active GRUB menu is: /rpool/boot/grub/menu.lst
default 0
timeout 30
0 OpenSolaris Development snv_134
```


כדי לשנות תכונה ב-Grub יש להשתמש בתת הפקודה set-menu

```
#bootadm set-menu [-R altroot] key=value
```

לדוגמא:

```
#pfexec bootadm set-menu timeout=5
```

התאוששות מהגדרות משובשות

במקרה ולא הצלחתם להיכנס לשולחן העבודה עם הגדרות המשתמש, ניתן להתאושש ממצב של הגדרות משובשות על ידי עליה מ-Live CD.

יש לעלות מ-Live CD של אופןסולריס לבחור בהגדרות ברירת מחדל עד לקבלת שולחן עבודה לפתוח מסוף ולהקליד את השורות הבאות:

```
#pfexec zpool import -f rpool
#mkdir /tmp/mnt
#pfexec set mountpoint=/tmp/mnt rpool/ROOT/opensolaris
#pfexec zfs mount rpool/ROOT/opensolaris
#pfexec nano /tmp/mnt/etc/shadow (find user ilan and delete the second field
between :)
#cd /
#pfexec zfs unmount rpool/ROOT/opensolaris
#pfexec zfs set mountpoint=/ rpool/ROOT/opensolaris
#pfexec zpool export rpool
```

ניהול שירותי מערכת הפעלה

נושאי לימוד:

- מנהל השירותים
- ניהול שירותים בממשק הפקודה
- יצירת שירות חדש – רשות

בפרק זה יילמדו התכנים הבאים:
מהי התשתית לניהול שירותים (Service Management Facility) בסביבת סולריס. כיצד לנהל שירותים (להפעיל או להפסיק שירות) על ידי שימוש בחלון מנהל השירותים או על ידי הפקודות svcs שמראה את כל השירותים והמצב שלהם במערכת (פעיל/לא פעיל/דורש תחזוקה/מאופשר/לא מאופשר). התלמיד יבין כיצד ניתן לראות שירותים ותלויות שלהם. ניתן לראות אילו שירותים תלויים בשירות שלי ובאילו שירותים השירות שלי תלוי. כמו כן התלמיד ידע כיצד ליצור שירות חדש ולהכניס אותו לתוך מאגר השירותים.

מנהל השירותים

על מנת להפעיל את מנהל השירותים:

System → Administration → Services

ניהול שירות בממשק הפקודה

הפקודה *svcs*

הפקודה *svcs* מאפשרת צפיה בכל השירותים שמוגדרים ופעילים במערכת.

האופציה *-a* מאפשרת לצפות בכל השירותים שמוגדרים במערכת (גם אלו שלא פעילים).

לדוגמא:

```
svcs [-a]
```

-a - על מנת לראות את כל השירותים (כולל הלא פעילים).

```
svcs dns/client
```

הפקודה *svcadm*

הפקודה *svcadm* מאפשרת ניהול שירותים על ידי שימוש בתתי הפקודות הבאים:

```
svcadm enable [-rst] <service> ... - enable and online service(s)
svcadm disable [-st] <service> ... - disable and offline service(s)
svcadm restart <service> ... - restart specified service(s)
svcadm refresh <service> ... - re-read service configuration
svcadm mark [-It] <state> <service> ... - set maintenance state
svcadm clear <service> ... - clear maintenance state
svcadm milestone [-d] <milestone> - advance to a service milestone
```

הפעלת שירות על ידי תת הפקודה *enable*

עצירת שירות על ידי תת הפקודה *disable*

עצירה והפעלה על ידי תת הפקודה *restart*

קריאה מחדש של קבצי קונפיגורציה של השירות על ידי תת הפקודה *refresh*

מחיקת מצב *maintenance* על ידי הפקודה *clear*.

מעבר לרמת ריצה (*milestone*) על ידי תת הפקודה *milestone*.

האופציה *-t* מאפשרת ביצוע פעולה בצורה זמנית. אם האופציה *-t* לא מופיעה הפקודה

תתבצע בצורה קבועה (תקף גם אחרי *reboot*).

רמות ריצה

ניתן לראות מהי רמת הריצה הנוכחית על ידי הפקודה -

who -r

svc:/milestone/single-user:default	=	s/1 – רמת ריצה תחזוקה ללא חלונות
svc:/milestone/multi-user:default	=	2 – רמת ריצה עם חלונות ללא שירותי שרת
svc:/milestone/multi-user-server:default	=	3 – רמת ריצה עם חלונות ועם שירותי שרת

רמת ריצה 5 - הינה כיבוי של המחשב
רמת ריצה 6 – אתחול מחדש

ניתן לעבור בין רמות ריצה בשתי צורות:

על ידי הפקודה init לדוגמא מעבר לרמת ריצה 2:

init 2

על ידי הפקודה svcadm milestone

svcadm milestone multi-user-server

הגדרת משתמשים וקבוצות

נושאי לימוד:

- שימוש בחלון המשתמשים והקבוצות
- קבצי הגדרות לניהול משתמשים וקבוצות
- שימוש בשורת הפקודה
 - הוספת קבוצה
 - מחיקת קבוצה
 - הוספת משתמש
 - מחיקת משתמש
 - שינויי סיסמה למשתמש

בפרק זה יילמדו התכנים הבאים:
מהם קבצי המידע המאכלסים את המשתמשים (קובץ passwd) הסיסמאות (קובץ shadow) והקבוצות (קובץ group) במערכת ההפעלה. וכיצד מבנה של כל ערך בקבצים אלו.
כיצד לנהל משתמשים וקבוצות (הוספה/עדכון/מחיקת משתמש, הוספה/עדכון/מחיקת קבוצה) על ידי שימוש בחלון מנהל המשתמשים והקבוצות או על ידי הפקודות useradd/usermod/userdel לניהול משתמש והפקודות groupadd/groupmod/groupdel לניהול קבוצות במערכת.

שימוש בחלון המשתמשים והקבוצות

System → Administration → Users and Groups

קבצי הגדרות לניהול משתמשים וקבוצות

קבצי ההגדרות לניהול משתמשים וקבוצות בנויים כמסד נתונים טקסטואלי. כל שדה בקבצי ההגדרות מופרד מהשדה הבא על ידי התו נקודותיים (:).

קובץ הגדרות לניהול קבוצות – /etc/group
כל שורה בקובץ הגדרות בנויה בפורמט הבא:

חברים:מספר מזהה: שם קבוצה
root::0:
other::1:root
bin::2:root,daemon

קובץ הגדרות לניהול משתמשים – /etc/passwd
כל שורה בקובץ הגדרות בנויה בפורמט הבא:

מעטפת: ספרית בית: שם אמיתי: קבוצה ראשית:מספר מזהה:לא בשימוש:שם משתמש
root:x:0:0:Super-User:/root:/usr/bin/bash
daemon:x:1:1:/:
bin:x:2:2:/:usr/bin:
nobody4:x:65534:65534:SunOS 4.x NFS Anonymous Access User:/:
haim:x:101:10:Haim Tzadok:/home/haim:/bin/bash

קובץ הגדרות לניהול סיסמאות וגילול(קביעת גיל) סיסמאות – /etc/shadow
כל שורה בנויה בפורמט הבא:

סיסמא מוצפנת:שם משתמש
root:\$5\$xWlv2bUl\$op8YDIEoPztueJtqGZ8igczNfoKTbn4ymyFWyBF9J1/:14818:::::::
daemon:NP:6445:::::::

שימוש בשורת הפקודה**ניהול קבוצות**

הוספת קבוצה

לדוגמא:

groupadd [-g gid] <group-name>

pfexec groupadd -g 5000 eng

מחיקת קבוצה

לדוגמא:

groupdel <group-name>

groupdel eng

שינוי הגדרות לקבוצה

groupmod <group-name>

ניהול משתמשים

הוספת משתמשים

useradd <-g primary-group-name> [-G group,group,...] [-u uid] [-d /export/home/<login-name>] [-m] <login-name>

-g - הקבוצה ראשית

-G - קבוצות משניות

-u - מספר מזהה למשתמש

-d - ספריית הבית

-m - ליצור את ספריית הבית

pfexec useradd -g eng -u 6001 -d /export/home/eli -m eli

מחיקת משתמשים

pfexec userdel [-r] <login-name>

שינוי הגדרות משתמשים

pfexec usermod [-u uid] [-g gid] [-G group,group,...] [-d /export/home/<login-name>] <login-name>

לדוגמא:

haim@opensolaris:/boot/grub\$ pfexec usermod -G mysql eli

צפיה בהגדרות של משתמשים

הפקודה id מציגה מידע על המשתמש.

דוגמא:

```
id [-a] <login-name>
```

האופציה -a - תציג גם את כל הקבוצות המשניות בהן אני חבר.

לדוגמא:

```
haim@opensolaris:/boot/grub$ id -a eli
uid=6001(eli) gid=6000(eng) groups=6000(eng)
```

החלפת משתמש במעטפת

הפקודה su מאפשרת להחליף זהות משתמש במעטפת. אם לא ניתן לפקודה את שם המשתמש אליו יש לעבור הפקודה תניח כי אנו רוצים להחליף זהות למשתמש root.

דוגמא:

```
su [-] [loginname]
```

אם הארגומנט הראשון הינו "-", המערכת תחליף את סביבת המשתמש (משתני הסביבה וקריאת קבצי אתחול) של המשתמש הישן בזה של המשתמש החדש.

לדוגמא אם נרצה להחליף את זהותנו במערכת להיות eli:

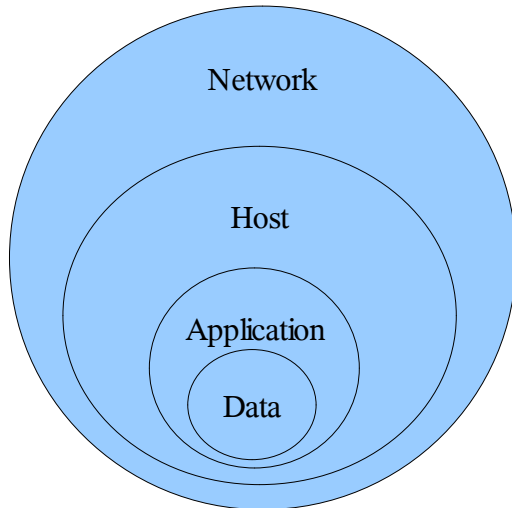
```
haim@opensolaris: ~$ su - eli
eli@opensolaris: ~$
```

אבטחה מקומית למערכת ההפעלה

- למה צריך אבטחה ?
- מהי הצפנה (Encryption) ומהו ערבול (hash)
- ניהול אבטחה מקומית למערכת ההפעלה

למה צריך אבטחה

אבטחה במערכת ההפעלה מאפשרת מניעת גישה למידע המאוחסן למשאבי מערכת ההפעלה ואפליקציות שרצות על מערכת ההפעלה. מניעת גישה למחשב או שימוש שאינו מורשה. ופעילות מאובטחת ברשת.



אבטחה בנויה בעיקר מן המנגנונים הבאים:

- גישה
- בקרה
- הצפנה ופענוח

1. גישה

1. התחברות – שימוש בשמות משתמשים עם יכולות מצומצמות וסמכויות מוגדרות RBAC – ידובר בהמשך. גיול סיסמאות, אכיפת מדיניות החלפת סיסמאות, אכיפת מדיניות יצירת סיסמאות (אלפא נומרי, אורך סיסמא, שימוש במילים שאינם מילות מילון וכו'). חסימת משתמש לאחר מספר נסיונות התחברות שנכשלו
ראו קובץ - /etc/default/passwd
2. גישה למשאבי זכרון, מעבד – הגדרת הרשאות לתהליכים ומידע בתוך תהליך המשתמשים שמריצים את התהליכים אינם root. או שמוגבלת לתהליך גישה והרשאה לבצע פעולות כלשהן.
3. גישה לרשת – מנגנוני זיהוי משתמשים ברשת – על ידי שם משתמש וסיסמא על ידי חתימה דיגיטלית כפי שקורה ב- ssh או במנגנונים כגון: kerberos.
4. גישה למערכת הקבצים – הרשאות ובעלויות ומתן סמכויות לניהול קבוצות מידע (data-sets) במאגר האחסון. מידור משתמשים על ידי מתן הרשאות מתאימות לקבצים. לדוגמא למצוא את כל הקבצים שהם world writable על ידי שימוש ב-
find

Find / -perm=777

2. הצפנה ופענוח

1. התחברות מאובטחת – שימוש בחתימה דיגיטלית (או טביעת אצבע) כפי שקיים ב- ssh עם fingerprint ומנגנוני זיהוי נוספים שאינם טקסטואליים.
2. ריצה מאובטחת- שימוש במנגנונים בטוחים בעת יצירת האפליקציה כגון: בדיקת גלישה מחוצצים, בדיקת קלט לא בטוח – נושאים מתקדמים בתכנות. לא בחומר שלנו.

3. רשת מאובטחת – מנגנונים להצפנה ופענוח בעת מעבר נתונים ברשת. אסור להשתמש בפרוטוקולים שאינם מוצפנים כגון: stmp, telnet, pop3 וכו'. חובה להשתמש ב-ssh להתחברות בין מחשבים. איסור להעביר סיסמאות ומידע רגיש ברשת ללא הצפנה.
הפקודה snoop מאפשרת להריץ סניפר על הרשת.
4. מערכת קבצים מאובטחת- הצפנת מידע על קבצים וספריות על ידי שימוש בהצפנה. (עתיד להיות קיים ב-ZFS).

3. בקרה

1. התחברות – פיקוח על מי מחובר (הפקודה who), פיקוח על תהליך ההתחברות ושימוש בקבצי יומן לתיעוד גישה. ניטור התחברויות שהצליחו או כשלו. קובץ /var/adm/sulog , או /var/adm/loginlog כדאי לבצע man sulog או man loginlog. הפקודה last מספקת תיעוד על המערכת.
2. בקרת ריצה – כלי תיעוד להרצת פקודות ותהליכים במערכת, הפסקת תהליכים שלא נצרכים.
3. בקרת גישה לרשת – תיעוד גישה לשירותי רשת, שימוש במנגנוני בקרת גישה כגון TCP Wrappers , חומת אש (IPF Firewall).
4. בקרת גישה למערכת הקבצים – תיעוד פעולות על מערכת הקבצים לדוגמא שימוש ב- zpool history.

הצפנה או ערבול (Hashing or Encryption)

הצפנה הנה מצב בו לוקחים מידע ומעבירים אותו דרך אלגוריתם שמשנה את המידע כך שניתן לחזור אליו עם אלגוריתם פענוח.

ערבול (hash) הנו מצב בו לקוחים מידע ומערבלים אותו דרך אלגוריתם שמשנה את המידע כך שלא קיים אלגוריתם פענוח.

שימושים

ערבול -

ניתן לבצע ערבול על מידע על מנת לבדוק תקינות שלו לדוגמא שימוש ב-md5 כאמצעי לבקרת תקינות קובץ. רוב יצרני התוכנה מספקים עם הפקודה את הערבול בצורת md5 על מנת לאמת כי התוכנה שהורדה אכן מגיעה מהיצרן ולא עברה שינויים. לדוגמא בדיקת חותמת md5 על ידי שימוש בפקודה digest:

```
digest -a md5 <file-name>
```

ניתן להשתמש גם בפקודה md5sum על ידי התקנת החבילה המתאימה המכילה את אותה פקודה.

```
md5sum <file-name>
```

הצפנה -

ניתן לבצע הצפנה למידע על ידי שימוש בפקודות חיצוניות כגון:

```
encrypt <file-name> - פקודה ליצירת קובץ מוצפן
```

```
decrypt <file-name> - פקודה לפתיחת קובץ מוצפן
```

נספח א - תוכנות מעניינות לשימוש מקומי

התקנת GIMP מאפשרת עריכה ועיבוד של קבצי תמונה בדומה ל- photoshop .
על מנת להתקין יש לבצע:

```
pfexec pkg install image/editor/gimp
```

התקנת Apache

Apache הוא השרת אינטרנט הנפוץ בעולם. על מנת להתקין יש לבצע:
pfexec pkg install web/server/apache-22 system/management/visual-panels/panel-apache

התקנת Adobe reader

על מנת להשתמש ב- Adobe reader, יש להוריד את תוכנת Adobe reader מהאתר של חברת
Adobe – www.adobe.com . ולהתקין את הקובץ לאחר ההורדה על ידי הרצה שלו.
pfexec ./Adobe*.run
על מנת לקבל את ערכי ברירת המחדל, יש להקיש enter בכל שאלה שתישאל, עד לסיום
ההתקנה.

התקנת flash-player

על מנת להשתמש ב- Adobe flash player, יש להוריד את תוכנת Adobe flash-player מהאתר
של חברת www.adobe.com – Adobe . ולפתוח את הקובץ ארכיב על ידי שימוש במנהל
הארכיב ולחלץ את הקובץ flashplayer.so לספרייה הנוכחית.

לאחר חילוץ הקובץ יש לבצע העתקה של הקובץ לספריית plugins הראשית של firefox .
ולאתחל את firefox .

```
pfexec cp libflashplayer.so /usr/lib/firefox/plugins/
```