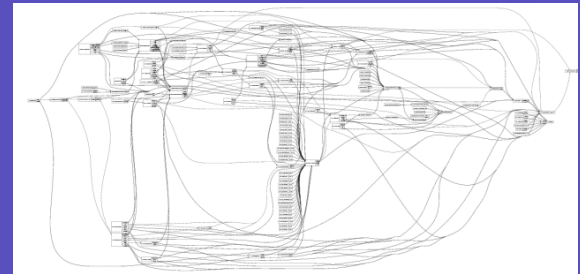


smf(5): Service Management Facility

Cserép János

Technical Specialist

Client Solutions/SW Practice



Predictive Self-Healing

Solaris Fault Manager
Solaris Service Manager (smf)

Solaris Fault Manager

- Hardverhibák felismerése és diagnózisa, mely az érintett alkatrészek gyors izolációját és pontos hibajelentéseket eredményez. Erről most nem lesz szó...

smf(5)

- A szolgáltatások futásának biztosítása, hiba esetén “öngyógyítása”

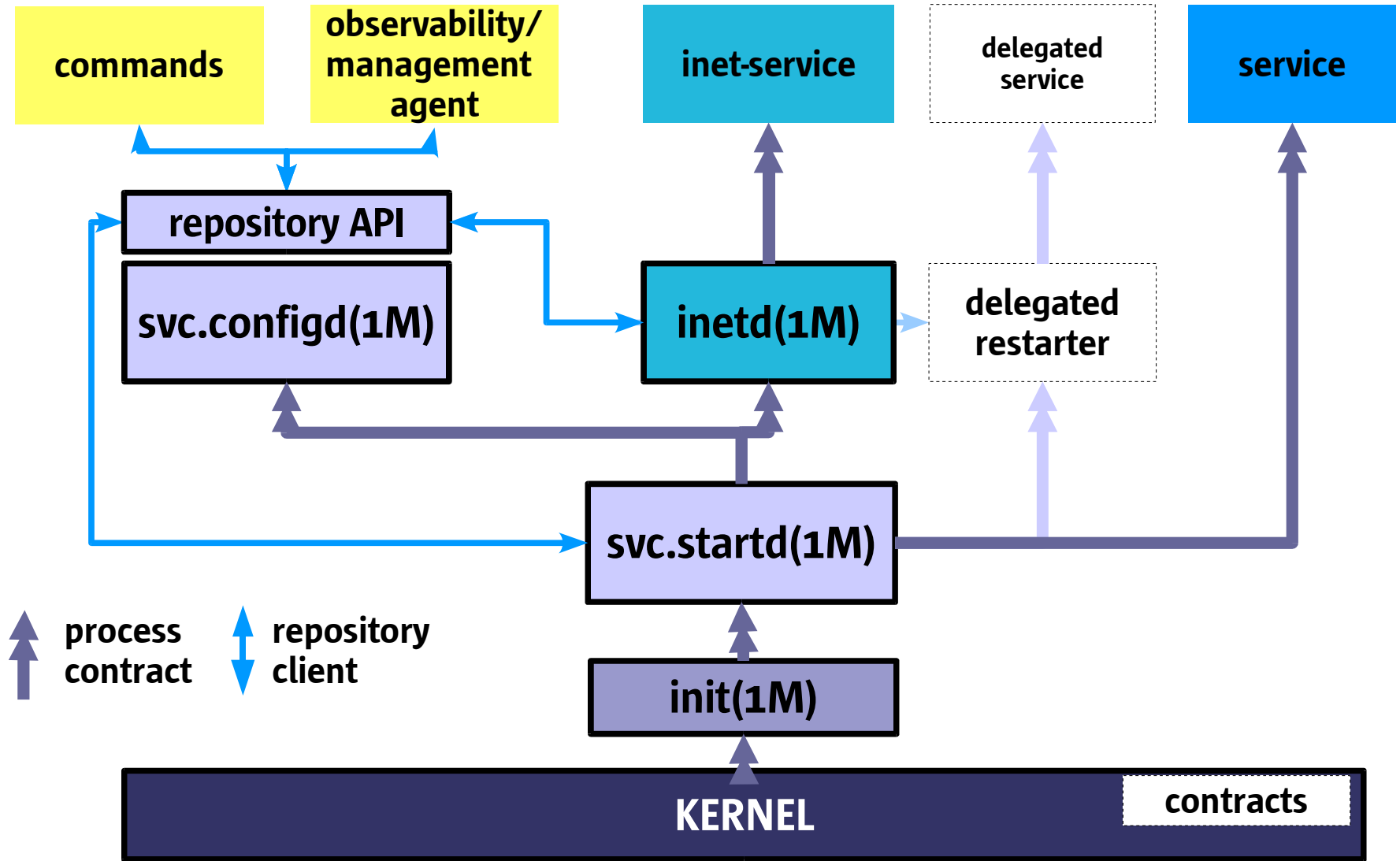
smf(5) Service Management Facility

- A szolgáltatás-indító scriptek, futási szintek és konfigurációs állományok új infrastruktúrája a Solaris 10-ben
 - Formalizált szolgáltatás függőségek
 - Szolgáltatás indítás, tiltás, újraindítás
 - Konfigurációs adatok tárolása
- Teljes kompatibilitás az /etc/rc* scriptekkel

smf(5) dióhéjban

- A rendszert egy “svc.startd” nevű démon felügyeli
- A démon betartatja a szolgáltatások közt definiált függőségeket, indít és leállít szolgáltatásokat igény és a rendszer hardverállapota szerint
- Az smf(5) konfigurációja egy adattárban (repository) tárolódik, melyet egy másik démon kezel (svc.configd)
- A rendszer tartalmaz egy vagy több delegált újraindító folyamatot, ami a szolgáltatások egy speciális részét felügyeli (pl inetd a hálózati szolgáltatásokhoz)

Az architektúra



Néhány következmény

- Sokkal csendesebb és gyorsabb boot folyamat

```
SunOS Release 5.10 Version Generic 64-bit
Copyright 1983-2005 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
Hostname: demobox
NIS domain name is testlab.example.com
checking ufs filesystems
/dev/rdisk/clt0d0s7: is logging.
```

```
demobox console login:
```

- Részletes boot: (boot -m verbose)

```
SunOS Release 5.10 Version Generic 64-bit
Copyright 1983-2005 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
[ network/loopback:default starting (loopback network interface) ]
[ system/filesystem/root:default starting (root file system mount) ]
[ network/pfil:default starting (packet filter) ]
[ network/physical:default starting (physical network interfaces) ]
...
```

Még néhány következmény

- A szolgáltatásfolyamatok immunisak a kill-re
 - Mert automatikusan újraindulnak
 - Az svcadm parancs szolgál az engedélyezésre/leállításra
- /etc/init.d és /etc/rc*.d könyvtárak, valamint az /etc/inittab fájl szinte üres a korábbi Solaris verziókhöz képest
- Szolgáltatás leállítása a tőle függő szolgáltatásokat is leállítja

Szolgáltatás

- A Solaris rendszer egy szoftver objektumának definíciója
- Egy szolgáltatás több példányban létezhet (pl webservert több porton)
- A példányok jól definiálható állapottal rendelkeznek [*uninitialized, offline, online, degraded, disabled, vagy maintenance*]

FMRI: Fault-Managed Resource Identifier

- Szolgáltatás:
`svc:/network/smtp:sendmail`
- Adatmező csoport és adatmező:
`svc:/network/smtp:sendmail/:properties/general/enabled`

Service Manifest

- A szolgáltatás XML formátumú definíciója
- A manifest file a szolgáltatás telepítésekor kerül a rendszerre (/var/svc/manifest)
- Automatikusan importálódik telepítéskor, boot-kor (az svc:/system/manifest-import által) és pkgadd futtatása esetén
- Importálás után nincs jelentősége (nem használja a rendszer)
- A Sun által szállított manifest fájlokat NE módosítsuk. Testreszabás a repository-ban történik (svccfg)

Legacy (rc.d) szolgáltatások

- Elindulnak és leállnak az `/etc/rc?.d` könyvtárban elfoglalt helyük szerint
- A hibáikat nem kezeli az smf, nincs újraindítás
- Az indító scriptek nevét listázza az `svcs` parancs `lrc:` előtaggal

Biztonság

- A szolgáltatások indítása és leállítása egyetlen ponton történik
- Alternatív profilok:
 - `svccfg apply /var/svcs/profile/generic_limited_net.xml`
- RBAC a szolgáltatáskonfigurációhoz(`smf_security(5)`):
 - `solaris.smf.modify`: szolgáltatás adatainak változtatása
 - `solaris.smf.manage`: újraindítás, leállítás, stb
- A manifest-ben egy `method_context` elem definiálásával megszabható, hogy egy szolgáltatás milyen jogosultsági szinten (`user`, `role`, stb) fusson

Kompatibilitás

- Meglévő SysV init scriptek (/etc/rc?.d) nem igényelnek változtatást, kivéve:
 - Ha egy Solaris által nyújtott szolgáltatás előtt kell elindulniuk
 - A konzolról kézzel bevitt adatokra van szükségük az induláshoz (erősen ellenjavalt)
- Ezen szolgáltatások “legacy”-ként látszódnak az smf parancsok kimenetében
- Az /etc/intittab bejegyzései továbbra is működnek
- Publikus konfigurációs állományok változatlanok, kivéve...

Az inetd.conf

- inetd.conf már nem központi konfigurációs állomány
- A Solarisba beépített inetd szolgáltatások az smf-et használják
- Előző Solaris verziókról történő upgrade-nél az `inetconv(1M)` parancs automatikusan konvertál
- Az `inetconv(1M)` parancsot kézzel kell futtatni inetd-t használó alkalmazások telepítése után, vagy az `inetd.conf` kézi módosításakor
- Boot közben figyelmeztető üzenet jön, ha van konvertálatlan bejegyzés

Delegált újraindító

- Egy szolgáltatáshalmazért felelős
- A szolgáltatási hibákat a saját modellje alapján kezeli
- Folyamatokhoz nem kötődő szolgáltatási modellt is nyújthat (pl J2EE alkalmazás szerver, mint újraindító, szolgáltatás mint J2EE alkalmazás)
- Gyakran extra funkcionalitást kínál megadott szolgáltatásokhoz (pl inetd)

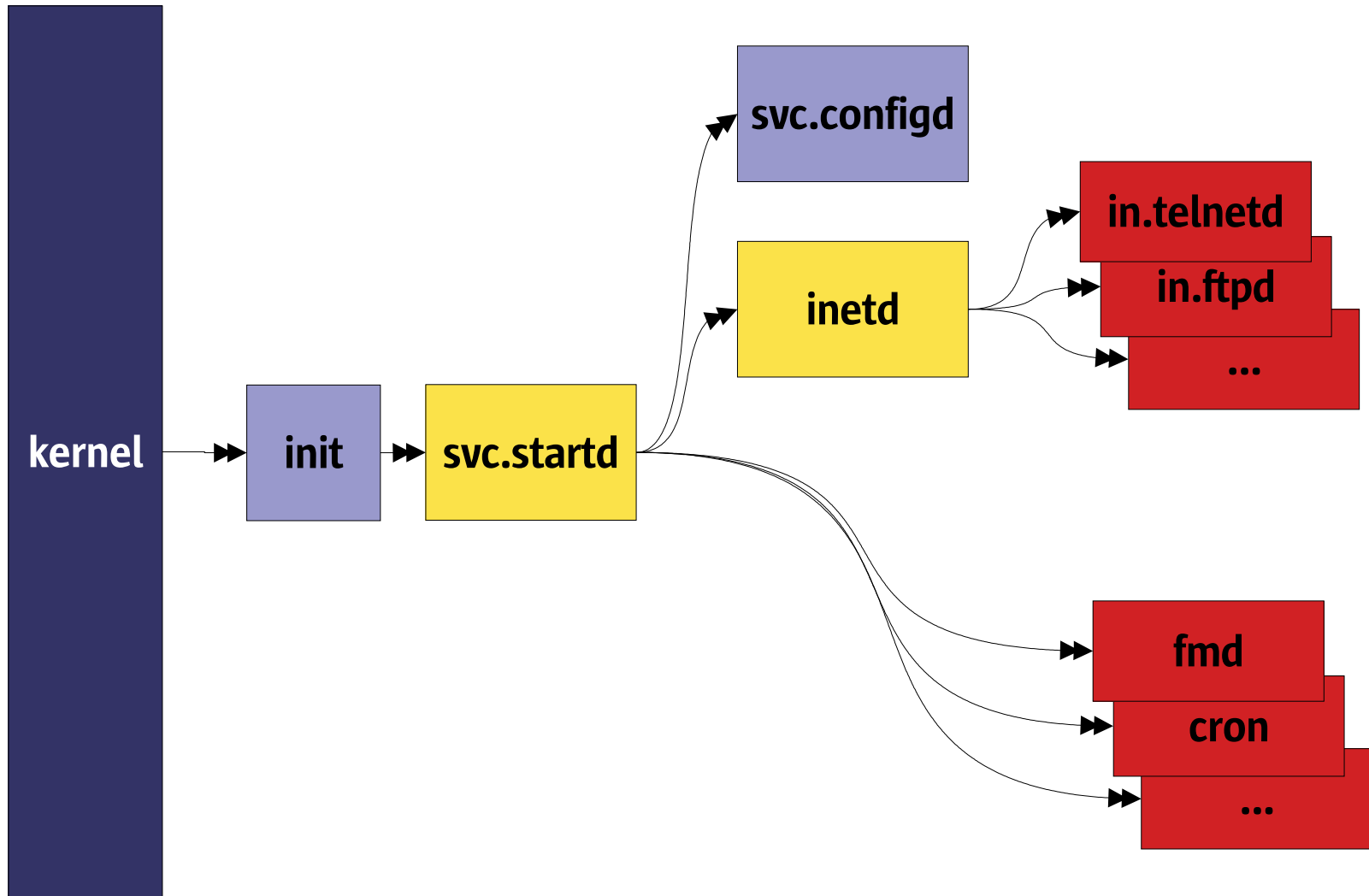
inetd

- Egy delegált újraindító a hálózati szolgáltatásokhoz
- Extra dolgok kezelése:
 - tcp_wrappers
 - max kapcsolatfelépítési ráta
- Bejövő kérések figyelése a szolgáltatások helyett
- A kezelt szolgáltatások állapotát is kezeli

Parancsok

- Általános:
 - `svcs(1)` szolgáltatáslista
 - `svcadm(1M)` adminisztrációs eszközök
 - `svccfg(1M)` a repository manipulálása
 - `svcprop(1)` adatmezők lekérdezése (scriptekből)
- `inetd(1M)` parancsok:
 - `inetadm(1M)` inetd szolgáltatások adminisztrációja
 - `inetconv(1M)` meglévő `inetd.conf` konverzója

Ki kit indít el/újra



Hibakeresés

- A szolgáltatási hibák naplózódnak (konzol, syslog)
- `svcs -x` kimenet
 - `/var/svc/log`
 - `/etc/svc/volatile`
- Újraindítás: `boot -m verbose`
- Indulás közben megálló rendszer esetén:
 - `boot -m milestone=none`
 - Belépés
 - `svcadm milestone all`

Visszaállítás

- Ha egy szolgáltatás nem működik, próbálkozzunk egy refresh-sel: `svcadm refresh <fmri>`
- Kövessük az `svcs -x` utasításait
- Visszatérés egy korábbi konfiguráció snapshot-hoz

```
$ svccfg -s system/cron:default
svc:/system/cron:default> listsnap
initial
last-import
previous
running
start
svc:/system/cron:default> revert start
svc:/system/cron:default> exit
$ svcadm refresh cron
$ svcadm restart cron
```

- Olvasnivaló: `/lib/svc/share/README`

Manifest írás

- A DTD jól dokumentált:
`/usr/share/lib/xml/dtd/service_bundle.dtd.1`
- Keressünk egy hasonló szolgáltatást a
“`/var/svc/manifest`” alatt
 - `system/utmp` egy egyszerű önálló demon
 - `system/coreadm` egy egyszerű konfigurációval rendelkező szolgáltatás
 - `network/telnet` egy egyszerű inet szolgáltatás

További információk

- Fejlesztői dokumentáció és quickstart guide:
<http://www.sun.com/bigadmin/content/selfheal/>
- Solaris System Administration Guide smf fejezetei:
<http://docs.sun.com/app/docs/doc/817-1985>
- A fejlesztők blogjai:
 - <http://blogs.sun.com/sch>
 - <http://blogs.sun.com/lianep>



smf(5): Service Management Facility

janos.cserep@sun.com



Handout, referencia slide-ok (angolul)

Commands: `svcs(1)`

- List enabled or all (-a) instances, sorted by state, time
- Explanations for errors/states (-x)
- Show dependencies (-d) and dependents (-D)
- Show member processes (-p), additional details (-v/-l)

```

$ svcs
STATE          STIME      FMRI
....
online         18:18:30   svc:/network/http:apache2
online         18:18:29   svc:/network/smtp:sendmail
....
$ svcs -p sendmail
STATE          STIME      FMRI
online         18:18:29   svc:/network/smtp:sendmail
                18:18:29   100180 sendmail
                18:18:29   100181 sendmail

$ svcs -d sendmail
STATE          STIME      FMRI
online         18:17:44   svc:/system/identity:domain
online         18:17:52   svc:/network/service:default
....

```

Commands: svcadm(1M)

- Enable, disable, refresh, restart service instances
- Mark in special states (maintenance)
- Synchronously wait for changes (-s)

```

$ grep lianep /etc/user_attr
lianep:::auths=solaris.smf.modify,solaris.smf.manage
$ svcs apache2
STATE          STIME          FMRI
-              ?              svc:/network/http:apache2
$ # create /etc/apache2/httpd.conf
$ svcadm enable apache2
STATE          STIME          FMRI
online         19:19:01      svc:/network/http:apache2
$ # edit /etc/apache2/httpd.conf
$ svcadm refresh apache2
$ svcs apache2
STATE          STIME          FMRI
online         19:19:33      svc:/network/http:apache2
$ svcadm disable apache2
$ svcs apache2
STATE          STIME          FMRI
disabled      19:20:07      svc:/network/http:apache2

```

Commands: `svcadm(1M)` actions

- **enable**: allow service to start once its dependencies are satisfied
- **disable**: stop service and do not allow it to start again
- **enable/disable -t**: enable/disable until the system is rebooted
- **refresh**: reload service configuration and run the refresh method (if any)
- **restart**: stop the service, then allow it to start once its dependencies are satisfied (no configuration change made)

Commands: `svcadm(1M)` milestone

- Milestone: A service which specifies a collection of dependencies which declare a specific state of system-readiness
- Major milestones, which are analogous to system run-levels can be reached directly from boot (`-m milestone=`), the standard `init` invocations, or via `svcadm: milestone/single-user, milestone/multi-user, milestone/multi-user-server`
- Transitions to milestones are implemented by temporary disable. `svcs` will show all enabled services that are not part of the milestone as temporarily disabled

Commands: svccfg(1M)

- Import, export manifests; apply, extract profiles
- Interactive mode for modifying properties

```

$ grep lianep /etc/user_attr
lianep:::auths=solaris.smf.modify,solaris.smf.manage
$ svccfg -v import /var/svc/manifest/network/http-apache2.xml
svccfg: Refreshed network/http:/apache2
svccfg: Successful import.
$ svccfg
svc:> select network/http:apache2
svc:/network/http:apache2> listprop
...
general                framework
general/enabled        boolean  false
...
start                   method
start/exec              astring  "/lib/svc/method/http-apache2 start"
start/timeout_seconds  count    60
start/type              astring  method
svc:/network/http:apache> editprop
[$EDITOR launches, allows direct editing of properties]
svc:/network/http:apache2> exit

$ svcadm refresh apache2      # read changed config
$ svcadm restart apache2     # restart with changed config

```

Commands: `svccprop(1)`

- List properties of services and instances
- Fetch in convenient forms for scripting
- View running or current props (`-c`), uncomposed (`-C`)

```
$ svccprop network/http:apache2
...
physical/entities fmri svc:/network/physical:default
physical/grouping astring optional_all
physical/restart_on astring error
physical/type astring service
start/exec astring /lib/svc/method/http-apache2\ start
start/timeout_seconds count 60
start/type astring method
stop/exec astring /lib/svc/method/http-apache2\ stop
stop/timeout_seconds count 60
stop/type astring method
restarter/auxiliary_state astring none
restarter/next_state astring none
restarter/state astring disabled
restarter/state_timestamp time 1102030556.737590000

$ svccprop -p enabled network/http:apache2
false
```

Commands: `inetadm(1M)`

- List services managed by `inetd`
- View (`-l`) and modify (`-m`) `inetd`-specific properties

```

$ inetadm
...
enabled    online          svc:/network/ftp:default
enabled    online          svc:/network/finger:default
disabled   disabled       svc:/network/login:eklogin
disabled   disabled       svc:/network/login:klogin
enabled    online          svc:/network/login:rlogin
disabled   disabled       svc:/network/rexec:default
enabled    online          svc:/network/shell:default

$ inetadm -l ftp
SCOPE      NAME=VALUE
           name="ftp"
           endpoint_type="stream"
           proto="tcp6"
           isrpc=FALSE
           wait=FALSE
           exec="/usr/sbin/in.ftpd -a"
           user="root"
...
default   tcp_wrappers=FALSE

```