Dokumentacja dotycząca rozproszonego dostępu do urządzeń w systemie OpenSSI

Projekt z przedmiotu Rozproszone Systemy Operacyjne

Andrzej Asztemborski
A.Asztemborski@elka.pw.edu.pl

16 czerwca 2005

Spis treści

1 Mechanizmy dostarczane przez oprogramowanie OpenSSI

1.1 Odzwierciedlenie urządzeń w systemie plików

Urządzenia w poszczególnych węzłach klastra są reprezentowane w lokalnych systemach plików poprzez wpisy w katalogu /dev (z wykorzystaniem możliwości dawanych przez device file system - devfs). Zasoby sprzętowe znajdujące się w innych aktywnych węzłach są wyszczególnione w podkatalogach numerycznych wewnątrz /dev. Pozwala to jawnie odwołać się do dowolnego urządzenia w każdym konkretnym węźle.

1.2 Urządzenia a migracja procesów

Przeniesienie biegu procesu na inny węzeł jest przezroczyste z punktu widzenia dostępu do urządzeń. Kontekst odwołań pozostaje niezmienny, proces nadal korzystać będzie ze sprzętu obecnego w punkcie pierwotnego rozpoczęcia wykonania.

1.3 Konsole wirtualne

Urządzenia konsol wirtualnych (pty) są utrzymywane jako zasób globalny, dzielony pomiędzy węzłami. Są one umieszczone w katalogu /cluster/dev/pts i przydzielane w miarę potrzeb różnym maszynom. Ich numery są unikalne wewnątrz całego klastra. Dowolna konsola ma przyporządkowany konkretny komputer ją obsługujący.

2 Realizacja odwołań do urządzeń

2.1 Problemy z jednoznacznością odwołania

Wewnątrz normalnego systemu Linux para numerów (major, minor), specyficznych jedynie dla rodzaju sprzętu, jednoznacznie specyfikuje wszystkie procedury jądra, odpowiedzialne za obsługę urządzenia. Ponieważ w systemie OpenSSI pliki takie są widoczne poprzez cluster file system wewnątrz globalnej przestrzeni nazw, w wielu różnych węzłach (a tym samym wewnątrz wielu katalogów /dev/numer_węzła/) znajdują się pliki - punkty wejściowe o tych samych numerach, zaś odnoszące się do takiego samego sprzętu występującego w kilku fizycznych komputerach. Przykładem niech będzie listing plików “fd0” w poszczególnych tego typu katalogach klastra.

˜> ls -l /dev/fd0
brw-rw—- 1 root floppy 2, 0 2002-03-14 22:56 /dev/fd0

˜> ls -l /dev/2/fd0
brw-rw—- 1 root floppy 2, 0 2002-03-14 22:56 /dev/2/fd0

2.2 Rozpoznawanie celu odwołania

Aby dostęp do pliku urządzenia zakończył się sukcesem, system plików musi zawierać pewne modyfikacje względem standardowej realizacji w systemach typu UNIX. Normalnym postępowaniem w wypadku napotkania sytuacji dostępu do pliku - urządzenia znakowego lub blokowego jest przekazanie żądania do jądra w celu wywołania odpowiedniej funkcji, określonej poprzez przypisanie do pary numerów (major, minor). Dzieje się tak niezależnie od tego, skąd pochodzi ów plik (które urządzenie składowania danych przechowuje wpis katalogowy go określający). Jedynym sposobem uniknięcia takiej interpretacji jest zamontowanie takiego fragmentu systemu plików z opcją “nodev”, która uniemożliwi jakiekolwiek skorzystanie z takiego odnośnika. W systemie OpenSSI konieczne jest zupełnie inne rozwiązanie. Niezbędne jest bowiem rozpoznanie fizycznego punktu docelowego odwołania. W tym wypadku oznacza to skierowanie wykonywanych operacji nie bezpośrednio do lokalnego jądra, lecz, po przeszukaniu drzewa punktów montowań, znalezienie urządzenia zawierającego katalog z wpisem definiującym cel operacji. Na tej podstawie program obsługi rozproszonego system plików (w tym wypadku klient) przekazuje dane dotyczące odwołania do programu (serwera) na maszynie, która udostępnia owo urządzenie. Tam dopiero odnajdowana jest (na podstawie numerów) odpowiednia funkcja, lokalnego już, jądra systemu i ona wykonuje żądany dostęp do sprzętu. W tym momencie osiągnięta jest przezroczystość tej funkcjonalności - dokładnie tak samo korzysta się zarówno z lokalnie zainstalowanych urządzeń, jak i tych dostępnych na fizycznie innych komputerach.

3 Praktyczne aspekty zagadnienia

3.1 Przykład wykorzystania rozproszonego dostępu

W celu prezentacji podstawowych funkcjonalności opisywanego systemu, pokażę przebieg realizacji polecenia otwarcia czytnika płyt CD na nielokalnym komputerze. Dla ustalenia uwagi przyjmijmy, że pracujemy na węźle numer 1. Spójrzmy więc, co dzieje się w przypadku wydania komendy:

eject /dev/cdrom

- 1. Program eject generuje żądanie dostępu do urządzenia /dev/cdrom.
- 2. Żądanie kierowane jest poprzez jądro do procedury obsługi rozproszonego systemu plików.
- 3. Po przejściu łańcucha rozwiązania nazwy pliku (wraz z ewentualnymi dowiązaniami symbolicznymi występującymi po drodze) procedura rozpoznaje element docelowy - wpis katalogowy reprezentujący urządzenie znakowe.
- 4. Na podstawie tablicy montowań (/proc/mounts) odnajdowane jest miejsce przechowywania katalogu ze znalezionym elementem.
- 5. Okazuje się, że jest to lokalny system plików devfs, co jednoznacznie ustala przekierowanie wywołania do właściwej procedury lokalnego jądra systemu.

Natomiast przebieg wykonania tego polecenia z innym argumentem będzie wyglądał tak:

eject /dev/2/cdrom

- 1. Program eject generuje żądanie dostępu do urządzenia /dev/cdrom.
- 2. Żądanie kierowane jest poprzez jądro do procedury obsługi rozproszonego systemu plików.
- 3. Po przejściu łańcucha rozwiązania nazwy pliku (wraz z ewentualnymi dowiązaniami symbolicznymi występującymi po drodze) procedura rozpoznaje element docelowy - wpis katalogowy reprezentujący urządzenie znakowe.
- 4. Na podstawie tablicy montowań (/proc/mounts) odnajdowane jest miejsce przechowywania katalogu ze znalezionym elementem.
- 5. Okazuje się, że jest to zdalne urządzenie zamontowane jako część struktury cluster file system.
- 6. Procedura inicjuje transakcję sieciową z komputerem, do którego przynależy ta część przestrzeni katalogów.
- 7. Tam połączenie jest odbierane poprzez serwer udostępniania plików i następnie rozpoznawane jako zdalny dostęp do urządzenia.

- 8. Na tej podstawie uruchamiana jest odpowiednia procedura jądra, powodująca otwarcie czytnika płyt CD, zaś wynik jej działania jest zwracany tą samą drogą, którą przybyło żądanie.

3.2 Napotkane problemy związane z rozproszonym dostępem do urządzeń

O ile korzystanie z prostych funkcjonalności urządzeń znakowych lub blokowych nie napotykało na szczególne trudności, o tyle bardziej zaawansowane zastosowania (związane z montowaniem zdalnych źródeł danych) było nieco bardziej kłopotliwe. Przyczyn problemu szukać należy w implementacji programu lub funkcji systemowej mount. Mianowicie przy tej operacji sprawdzana jest "lokalność" montowanego urządzenia, co, w wypadku użytkowania rozproszonego systemu plików w komputerze nie będącym częścią klastra, uniemożliwia pomyłki dotyczące niejednoznaczności przyporządkowania sprzęt - para numerów (major, minor) dla wielu komputerów. Jednak w wypadku zastosowania rozwiązania takiego, jak OpenSSI, to postępowanie utrudnia korzystanie z przezroczystosci zapewnianej przez warstwę usług cluster file system. Przejawia się to następująco (przy pracy w węźle numer 1):

˜> mount -t vfat /dev/2/fd0 /media/floppy2
/dev/2/fd0 jest obiektem zdalnym zamontowanym przez NFS

W przypadku prostych urządzeń (np. stacji dyskietek) można zastosować nieeleganckie rozwiązanie "oszukujące" polecenie mount, wykorzystując pseudo-urządzenia blokowe loop:

˜> mount -t vfat /dev/2/fd0 /media/floppy2 -o loop

Wtedy /dev/2/fd0 zostanie potraktowane jak zwykły plik. Operacje na nim będą przechodzić poprzez jądrowy interfejs loop, a dopiero później trafią do systemu plików, który prawidłowo przekieruje je do odpowiedniej stacji dyskietek na właściwym komputerze. Niestety w wypadku urządzeń o bardziej skomplikowanej obsłudze (np. czytniki CD-ROM) ta technika nie zdaje egzaminu. Najprawdopodobniej interfejs loop nie obsługuje właściwych dla tego sprzętu komend związanych z funkcją ioctl i to uniemożliwia tego typu "sztuczki". Zapewne z tego powodu wynikały również usterki zaobserwowane przy tym wywołaniu w różnych okolicznościach (niedeterministyczne działanie, błędy typu naruszenie ochrony pamięci, itp.). Rozwiązaniem tego problemu może być przeanalizowanie i poprawienie (zgodnie z wymaganiami OpenSSI oraz cluster file system) wywołania systemowego mount lub programu tę funkcję realizującego na poziomie użytkownika.

Literatura

[ope] *Introduction to the SSI Cluster*.