

Linux, stuurprogramma's en kernelprogrammering

Praktijkcursus van 4 dags - 28h

Ref : LDI - Prijs 2025 : € 2 450 excl. BTW

Deze cursus stelt u in staat om de ontwikkeling van robuuste stuurprogramma's voor randapparatuur, aangepast aan de verschillende Linux-distributies, onder de knie te krijgen. U leert over de verschillende soorten randapparatuur, geheugenbeheer, netwerkprotocolimplementatie en USB-randapparatuur.

PEDAGOGISCHE DOELSTELLINGEN

Na afloop van de opleiding kan de cursist:

De ontwikkeling van stuurprogramma's voor randapparatuur onder de knie krijgen

Een gedetailleerd begrip van de interne werking van de kern

Weten hoe u nieuwe elementen ontwikkelt en integreert in de Linux-kernel

Een randstuurprogramma schrijven in teken- of blokmodus

HANDS-ON WORK

De talrijke oefeningen en progressieve casestudies worden uitgevoerd op een netwerk van Linux-servers. Alle programma's die tijdens de praktijksessies worden ontwikkeld, zijn beschikbaar in de vorm van skeletten die deelnemers zelf kunnen voltooien.

HET PROGRAMMA

laatste update: 02/2025

1) Kernpresentatie

- Overzicht van het systeem en de rol van de kernel.
- Referentiesites.
- Specifieke kenmerken van de 3.x en 4.x kernels.
- Kernelontwikkelingscycli, patches.
- Bedrijfsmodus (supervisor en gebruiker). Systeemoproepen.
- Organisatie van bronnen (Include/linux, Arch, Kernel, Documentatie, enz.).
- Principe van kernel- en modulecompilatie.
- Afhankelijkheden en symbolen.
- Export van symbolen.
- Kernel laden (ondersteuning, argumenten, enz.).

Een 3.x kernel compileren en installeren.

2) Tools die u kunt gebruiken

- Ontwikkelgereedschappen (Gcc, Kbuild, Kconfig en Makefile, enz.).
- Debug-gereedschappen (GDB, KGDB, ftrace, enz.).
- Debugomgeving (Linux Trace Toolkit, enz.).
- Versiebeheerprogramma (Git, etc.).
- Systeemaanroepen traceren (ptrace...).

Installeer alle hulpmiddelen en bronnen om een module voor de kernel te genereren. Het systeem configureren om modules automatisch te laden tijdens het opstarten. Eenvoudige modules schrijven en testen.

3) Beheer en planning van threads

- De verschillende soorten randapparatuur.
- Kernel besturingscontexten. Bescherming van globale variabelen.

DEELNEMERS

Linux/Unix-ontwikkelaars.

VOORAFGAANDE VEREISTEN

Goede kennis van Linux/Unix en C-programmeren.

VAARDIGHEDEN VAN DE CURSUSLEIDER

De deskundigen die de cursus leiden zijn specialisten op het betreffende vakgebied. Zij werden geselecteerd door onze pedagogische teams zowel om hun vakken als hun pedagogische vaardigheden voor elke cursus die zij geven. Zij hebben minstens vijf tot tien jaar ervaring in hun vakgebied en oefenen of oefenden verantwoordelijke bedrijfsfuncties uit.

BEOORDELINGSMODALITEITEN

De cursusleider beoordeelt de pedagogische vooruitgang van de deelnemer gedurende de gehele cursus aan de hand van meerkeuzevragen, praktijksituaties, praktische opdrachten, ... De deelnemer legt ook van tevoren en naderhand een test af ter bevestiging van de verworven kennis.

PEDAGOGISCHE EN TECHNISCHE MIDDELEN

- De gebruikte pedagogische middelen en cursusmethoden zijn voornamelijk: audiovisuele hulpmiddelen, documentatie en cursusmateriaal, praktische oefeningen en correcties van de oefeningen voor praktijkstages, casestudies of reële voorbeelden voor de seminars.
- Na afloop van de stages of seminars verstrekt ORSYS de deelnemers een evaluatievragenlijst over de cursus die vervolgens door onze pedagogische teams wordt geanalyseerd.
- Na afloop van de cursus wordt een presentielijst per halve dag verstrekt, evenals een verklaring van de afronding van de cursus indien de stagiair alle sessies heeft bijgewoond.

TOEGANGSMODALITEITEN EN -TERMIJNEN

De inschrijving dient 24 uur voor aanvang van de cursus plaatsgevonden te hebben.

TOEGANKELIJKHEID VOOR MINDervalIDEN

Is voor u speciale toegankelijkheid vereist? Neem contact op met mevr. FOSSE, contactpersoon voor mindervaliden, via het adres psh-accueil@ORSYS.fr om uw verzoek en de haalbaarheid daarvan zo goed mogelijk te bestuderen.

- Thread-weergave (status, task_stru structuur, thread_info, enz.).
- Draden, uitvoeringscontext.
- De Linux scheduler en pre-emption.
- Een kernel thread aanmaken (kthread_create, wakeup_process...).

Maak een module die een kernel thread aanmaakt tijdens het invoegen en ontlaadt tijdens rmmmod. Schrijf een hoge-precisie event timestamping module. Schrijf een module die informatie geeft over de interne structuren van processen.

4) Geheugen-, tijd- en procesbeheer

- Geheugenorganisatie voor UMA- en NUMA-architecturen.
- Gebruikers- en kerneladresruimte.
- Pagina's op aanvraag beheren (demand paging).
- Geheugentoewijzingen, buddy allocator, kmalloc, slabs en geheugenpools.
- Geheugentoegangsbeheer (caches en MMU).
- Problemen in verband met geheugenoverreservering.
- Geheugenbeheer op x86 en ARM, gebruik van Hugepages.
- Optimalisatie van systeemaanroepen (IAPX32, VDSO).
- Synchronisatie en wachten in de kernel, waitqueues, mutexen en completies.
- Tikken en jiffies in Linux.
- Real Time Clock (RTC), implementatie van timers.
- Hoge-resolutie timerinterface, stempels.
- Kernelspecifieke gereedschappen, gekoppelde lijsten, kfifo en container_of.
- De kernelinterface met /proc via procfs.

Timers en tijdstempels gebruiken. Toegang tot procfs implementeren. Geheugentoewijzing in de kernel implementeren en optimaliseren met slabs.

5) Apparaat in tekenmodus

- Stuurprogramma's voor karakterapparaten schrijven.
 - VFS (virtueel bestandssysteem).
 - Methoden voor karakterrandaapparatuur.
 - Beheer van DMA onderbrekingen en toegang tot hardware.
 - Registratie en optimalisatie van het stuurprogramma voor tekenapparaten.
- Geleidelijk schrijven van een character-mode perifere driver. Implementeren van I/O synchronisatie tussen threads en met de interrupt routine. Implementeren van geheugentoewijzing.*

6) Linux stuurprogramma-raamwerk - sysfs

- Presentatie van het framework, kobject, kset en kref.
 - Stuurprogramma, apparaatstuurprogramma, bus en klasse-objecten.
 - Gebruik en generatie van attributen in sysfs.
 - Interface met hotplug-, match-, probe- en release-methoden.
 - Firmwarebeheer.
 - Energiebeheer, energiebeheermethoden.
- Een bus, een stuurprogramma en een apparaatstuurprogramma implementeren. Het karakterapparaatstuurprogramma aanpassen. Voorbeeld van gebruik van de interface.*

7) Blokmodus randapparatuur en bestandssystemen

- Principe van blokmodus randapparatuur. Registratie van stuurprogramma's.
 - Callback voor lezen en schrijven. Ondersteuning voor opmaak en geavanceerde bewerkingen.
 - Blok-I/O scheduler voor de kernel.
 - Ontwerp van bestandssystemen.
 - Registratie van een nieuw bestandssysteem.
- Voorbeeld van een compleet virtueel apparaatstuurprogramma. Voorbeeld van een aangepast bestandssysteem.*

8) Netwerkkinterfaces en protocollen

- Netwerkkinterfaces beheren onder Linux.
- Gebruik van skbuff.
- Netfilterhaken.
- Integratie van een protocol.

Voorbeeld van een netwerkstuurprogramma voor een virtueel apparaat. Implementatie van een netwerkprotocol.

9) Stuurprogramma's voor USB-randapparatuur

- Principe van USB-randapparatuur. Interface met de USB-kernmodule.
- Interactie van het apparaat met de Linux kernel.
- Bouw van een URB (USB Request Block).
- USB-gadgets.

Een USB-stuurprogramma registreren. Een stuurprogramma schrijven in isochrone modus.

DATA

Neem contact met ons op