

Case study: Pintos

Mål och syfte

I denna uppgift skall ni tillsammans i en grupp om 4 personer studera ett område i boken och jämföra hur det är implementerat i Pintos. Avsikten är att dels få kunskaper inför duggan, dels få ökad förståelse för Pintos inför kommande laborationer. Grupperna är konstruerade för att en i varje labpar skall få respektive del. Studietekniskt ingår detta i momentet LAB1 och räknas som en vanlig lab med labrapport som slutprodukt. (Det är knappast första laborationen där ni studerar ett fenomen på lab och skriver en labrapport, även om det kan kännas underligt när “fenomenet” är programkod.)

“SGG8” Refererar till boken “Operating System Concepts” av Silberschatz, Galvin, Gagne, 8th edition. Har du en äldre eller nyare upplaga är det bara leta upp motsvarande sektioner. “Pintos” refererar till Pintos källkoden ni installerat på lab, och tillhörande dokumentation.

Ni jobbar i grupp om 4 och fördelar arbetet som ni vill. Eftersom allt hänger samman är det förmodligen smart att alla ögnar igenom alla delar till en början för att sedan titta på var sin specifik del. Om alla läser var sin del är det förmodligen svårt att få ihop en helhet. Varje grupp skall studera anvisat delsystem¹. Antingen “Processer och trådar” eller “Öppna filer och filsystem” enligt separat anvisning. Grupperna kommer presenteras på kurshemsidan.

Slutresultatet skall vara en rapport som beskriver Pintos implementation med referenser till boken, samt kommentarer hur problemet kan lösas på ett bättre sätt (med referens till boken). Alla frågeställningar som anges skall besvaras. Rapporten skall vara minst 3 sidor lång och får vara max 5 sidor lång. Alla rapporter kommer publiceras på kurshemsidan så alla kan få insikt i båda delsystemen.

Rapporten skall skrivas enligt ACM:s rapportmall vad gäller stil och layout med några fränsteg som samtidigt listar det viktigaste:

- Språk är enkel och rättfram svenska.
- Pappersformat är A4.
- Vi går upp +2 punkter i textstorlek jämfört med mallen. Normaltexten blir alltså 12 punkter Times (New) Roman.
- Det skall finnas en inledning, egna relevanta rubriker, en förklarad figur (se frågeställningar), en sammanfattande slutsats och en referenslista (som blir ganska kort och förutsägbar).

Att mallen följts i alla detaljer kommer inte kontrolleras, men det får inte vara uppenbart att den **inte** följts. Vid tveksamhet, lägg hellre tid på innehållet.

Det är **inte** meningen att ni skall producera en punktlista med svar på frågorna, utan att ni skall beskriva hur Pintos fungerar och diskutera för och nackdelar med lösningarna i Pintos. Frågorna är bara en ledning vad som ska finnas med i diskussionen. Det ingår alltså i varje fråga att jämföra med alternativa metoder som beskrivs i boken.

¹Ett sammanträffande är att det jag för vardera uppgift bedömt vara mest intressant att läsa i bok och kod räkar landa på samma (!) antal sidor, och exakt (!) samma antal rader kod i C-filerna i båda projekten. Fullständigt otroligt!

Processer och trådar

Läsanvisningar: 32 sidor, 914 rader

- SGG8: Section 3.1 - Process concept (4 sidor)
- SGG8: Section 3.2 - Process scheduling (5 sidor)
- SGG8: Section 3.3 - Operations on processes (6 sidor)
- SGG8: Section 4.1 - Overview (4 sidor)
- SGG8: Section 4.2 - Multithreading models (2 sidor)
- SGG8: Section 5.3 - Scheduling algorithms (11 sidor)
- SGG8: Relevanta förkunskaper till ovan sektioner
- Pintos: Relevant dokumentation
- Pintos: threads/thread.[ch] (639 rader)
- Pintos: userprog/process.[ch] (275 rader)
- Pintos: Övriga filer relevanta för förståelsen.

Frågeställningar

- Boken tar upp begreppet “Process Control Block” (PCB). I vilken datastruktur håller Pintos reda på motsvarande information?
- Boken tar upp begreppet “Thread Control Block” (TCB). I vilken datastruktur håller Pintos reda på motsvarande information?
- Processer har i OS olika states och är placerade i köer, däribland ready-kön. Var finns ready-kön i Pintos och vilka funktioner i Pintos implementerar förflyttningar mellan köerna? Vilken förflyttning gör varje funktion? Illustrera detta med en förklarad figur. Hur tar man reda på vilken process som exekverar?
- Vilken kod är ansvarig för schemaläggning av processer i Pintos och vilken algoritm används?
- Jämför de funktioner som finns för att hantera processer i Pintos med vad som beskrivs i boken.
- Jämför de funktioner som finns för att hantera trådar i Pintos med vad som beskrivs i boken.
- Hur startas en process i Pintos jämfört med boken?
- Föreslå ett sätt att utöka Pintos så processer kan starta flera (user-)trådar som delar processens resurser med mappningen 1-1 (one-to-one model) mellan user-trådar och kerneltrådar. Ange hur datastrukturen för nya PCB och nya TCB skulle se ut.

Öppna filer och filsystem

Läsanvisningar: 32 sidor, 914 rader

- SGG8: Section 10.3 - Directory and disk structure (11 sidor)
- SGG8: Section 11.1 - File-system structure (3 sidor)
- SGG8: Section 11.2 - File-system implementation (6 sidor)
- SGG8: Section 11.3 - Directory implementation (1 sida)
- SGG8: Section 11.4 - Allocation methods (8 sidor)
- SGG8: Section 11.5 - Free-space management (3 sidor)
- SGG8: Relevanta förkunskaper till ovan sektioner
- Pintos: Relevant dokumentation
- Pintos: `filesystem/directory.[ch]` (236 rader)
- Pintos: `filesystem/file.[ch]` (140 rader)
- Pintos: `filesystem/filesys.[ch]` (114 rader)
- Pintos: `filesystem/free-map.[ch]` (88 rader)
- Pintos: `filesystem/inode.[ch]` (336 rader)
- Pintos: Övriga filer relevanta för förståelsen.

Frågeställningar

- Boken tar upp begreppet “File Control Block” (FCB). I vilken datastruktur håller Pintos reda på motsvarande information?
- I boken beskrivs “system-wide open file table” och “per-process file table”. Var hittar du motsvarande datastrukturer i Pintos? Vilka funktioner manipulerar dessa? Lagras tabellerna på disk? Illustrera tabellerna med en förklarad figur. Varför behövs dessa tabeller?
- Jämför begreppet “inode” i boken och i Pintos.
- Boken beskriver flera metoder för “free-space management”. Hur fungerar det i Pintos?
- Vilken av de allokeringsalgoritmerna som boken tar upp används i Pintos?
- Jämför katalogstruktur och filhantering i Pintos med hur det är beskrivet i boken.
- Hur hanterar Pintos borttagning av filer som är öppna?
- Föreslå hur Pintos katalog och filsystem kan förbättras så att filer kan utökas utan external fragmentation. Länkar skall kunna skapas. Indexed allocation skall användas. Ange hur datastrukturen för nya tabellerna och nya FCB eller inode skulle se ut och hur en viss filposition kan räknas fram.