LINKÖPINGS UNIVERSITET

TDIU16: Process- och operativsystemprogrammering

Säkra systemanropen

Klas Arvidsson, Daniel Thorén, Filip Strömbäck

INSTITUTIONEN FÖR DATAVETENSKAP

Höstterminen 2018

2020-03-06

1 Mål

Hittills har vi i stor utsträckning utgått från att de program som körs i Pintos inte anropar systemanropen på ett felaktigt sätt. Detta är såklart inte hållbart i ett riktigt operativsystem eftersom det är lätt hänt att ett program innehåller buggar, eller att vi kör ett program som vi inte litar på som kanske aktivt försöker hitta problem i operativsystemet som kan användas för att ta över systemet.

Målet med den här laborationen är alltså att se till att systemanropen är så pass robusta att systemet inte kraschar på grund av att de anropas med felaktiga parametrar, eller på något annat konstigt sätt.

2 Uppgift

I denna laboration ska säkra alla systemanrop så att ett användarprogram inte kan förstöra för OS eller andra användarprogram genom att skriva/läsa till minne som den inte har tillgång till. Mer information om hur detta fungerar och vad som behöver göras finns i Pintos-Wiki under rubriken Minne. Läs särskilt underrubrikerna ”Paging i pintos” och ”Accesskontroll”.

2.1 Uppgift

Verifiera att all indata till varje systemanrop är korrekt och inte kan orsaka säkerhetsbrister eller krasher i kernel. Om någon parameter är ogiltig eller konstig skall systemanropet misslyckas. Om det går att misslyckas genom att returnera en felkod (oftast -1) från systemanropen skall detta göras, annars dödas det användarprogram som gjorde det felaktiga systemanropet. Vid allvarliga fel, såsom att en ogiltig pekare gavs som parameter, skall processen alltid dödas. En process som dödades av kernel skall alltid avsluta med -1 som *exit_status*.

Speciellt uppmärksam måste du vara på alla instruktioner som använder användarprogrammets minne (läser eller skriver). Flera pekare dit existerar, se tidigare laborationer. Du kan återanvända kod från uppgiften ”Accesskontroll”, men du måste lägga till några kontroller, och du måste skicka in korrekt `pagedir` till `pagedir_get_page`.

2.2 Testa din implementation

Efter denna laboration ska alla automatiska tester i Pintos fungera. Se Pintos-Wiki under ”Automatiska tester” för information om hur de körs.

Observera att testerna måste köras **många** gånger för att vara någotsånär säker på att allt fungerar. Synkroniseringsfel uppstår inte vid varje körning, utan någon gång ibland, och på olika ställen. Prova följande script för att sluttesta. Hur många vändor klarar din implementation?

```
pintos-check-forever -a
```