Zakleszczenie w wieloprocesowym systemie poczty elektronicznej

Tomasz Sienkiewicz

- Prezentacja powstała na podstawie pracy „A CCS-based Investigation of Deadlock in a Multi-process Electronic Mail System”
- Autorem pracy jest Gordon Brebner

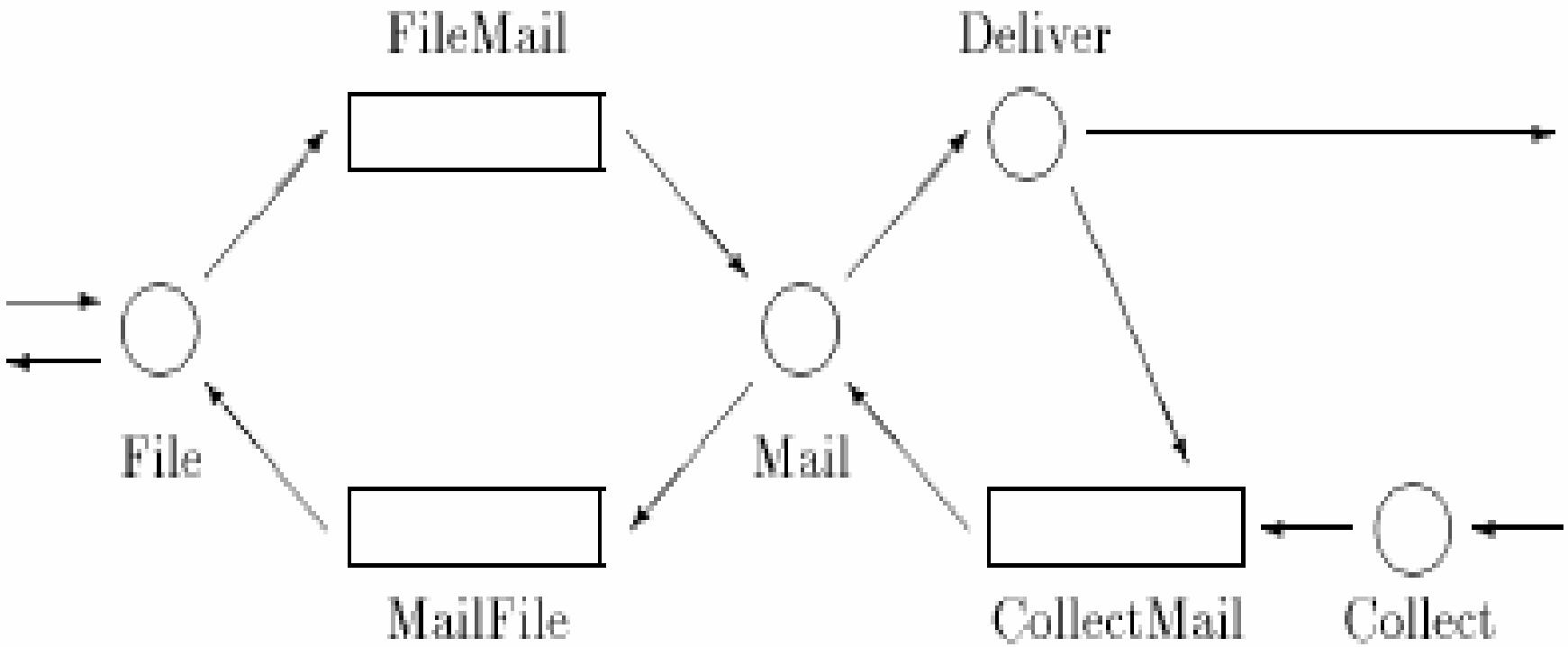
Wstęp

- System poczty elektronicznej na Uniwersytecie w Edynburgu
- Jeden z użytkowników ustawił przekazywanie poczty przychodzącej z „Dr Who Fan Club” do 20 osób na terenie całej Wielkiej Brytanii
- Pewnej nocy w krótkim odstępie czasu przychodzi 30 listów z „Dr Who Fan Club”
- Następuje awaria. Odbieranie i wysyłanie poczty jest nie możliwe, aż do restartu systemu

Struktura systemu

- Architektura wieloprocesowa:
  - Jeden proces obsługuje protokół X.25 i kontroluje fizyczne połączenie
  - Jeden proces obsługuje protokół XXX do zdalnego logowania
  - Jeden proces obsługuje transfer plików
  - Jeden proces obsługuje protokół poczty elektronicznej
- Komunikacja za pomocą pamięci dzielonej chronionej semaforami i bufory jednokierunkowe

Model CCS



Model CCS

- Cztery akcje zewnętrzne:
  - Odbieranie przesyłki z sieci
  - Wysyłanie przesyłki
  - Przekazywanie przesyłki dla użytkownika
  - Odbieranie przesyłki od użytkownika

Model CCS

$$
\begin{array}{l}
(\mathrm{Collect} \\
\quad \mid \mathrm{CollectMail} \\
\quad \mid \mathrm{Deliver} \\
\quad \mid \mathrm{Mail} \\
\quad \mid \mathrm{MailFile} \\
\quad \mid \mathrm{FileMail} \\
\quad \mid \mathrm{File}) \\
\setminus \{\mathrm{cm\_insert},\ \mathrm{cm\_remove},\ \mathrm{cm\_empty}, \\
\quad \mathrm{md\_start},\ \mathrm{md\_stop}, \\
\quad \mathrm{mf\_insert},\ \mathrm{mf\_remove},\ \mathrm{mf\_empty}, \\
\quad \mathrm{fm\_insert},\ \mathrm{fm\_remove},\ \mathrm{fm\_empty}\}
\end{array}
$$

Model CCS - bufory

$$\text{Buffer} \stackrel{\text{def}}{=} \text{insert} \,.\, \overline{\text{remove}} \,.\, \text{Buffer} \;+\; \overline{\text{empty}} \,.\, \text{Buffer}$$

$$\begin{aligned}
\text{CollectMail} &\stackrel{\text{def}}{=} \text{Buffer}\,[\text{cm_insert}/\text{insert},\ \text{cm_remove}/\text{remove},\\
&\qquad\qquad \text{cm_empty}/\text{empty}]\\
\text{MailFile} &\stackrel{\text{def}}{=} \text{Buffer}\,[\text{mf_insert}/\text{insert},\ \text{mf_remove}/\text{remove},\\
&\qquad\qquad \text{mf_empty}/\text{empty}]\\
\text{FileMail} &\stackrel{\text{def}}{=} \text{Buffer}\,[\text{fm_insert}/\text{insert},\ \text{fm_remove}/\text{remove},\\
&\qquad\qquad \text{fm_empty}/\text{empty}]
\end{aligned}$$

Model CCS – proces Collect

- W każdym momencie proces użytkownika może umieścić email w buforze CollectMail
- W modelu jest tylko jeden proces, który wkłada listy do bufora

$$\mathrm{Collect} \stackrel{\mathrm{def}}{=} \mathrm{letter\_posted} \,.\, \overline{\mathrm{cm\_insert}} \,.\, \mathrm{Collect}$$

Model CCS – proces Deliver

- Proces niedeterministycznie wykonuje jedną z dwóch akcji:
  - dostarcza przesyłkę użytkownikowi
  - przesyła email dalej – wkłada do CollectMail

$$\mathrm{Deliver} \stackrel{\mathrm{def}}{=} \mathrm{md\_start} \,.\, (\tau \,.\, \overline{\mathrm{letter\_deliver}} \,.\, \overline{\mathrm{md\_stop}} \,.\, \mathrm{Deliver} + \tau \,.\, \overline{\mathrm{cm\_insert}} \,.\, \overline{\mathrm{md\_stop}} \,.\, \mathrm{Deliver})$$

Model CCS – proces Mail

- Proces wykonuje po kolei następujące akcje:
  - Wyjmuje przesyłkę z bufora FileMail, jeżeli bufor jest niepusty
  - Niedeterministycznie albo przekazuje email do użytkownika albo generuje komunikat o błędzie i wkłada go do bufora MailFile
  - Przenosi wszystkie wiadomości z bufora CollectMail do MailFile

Model CCS – proces Mail

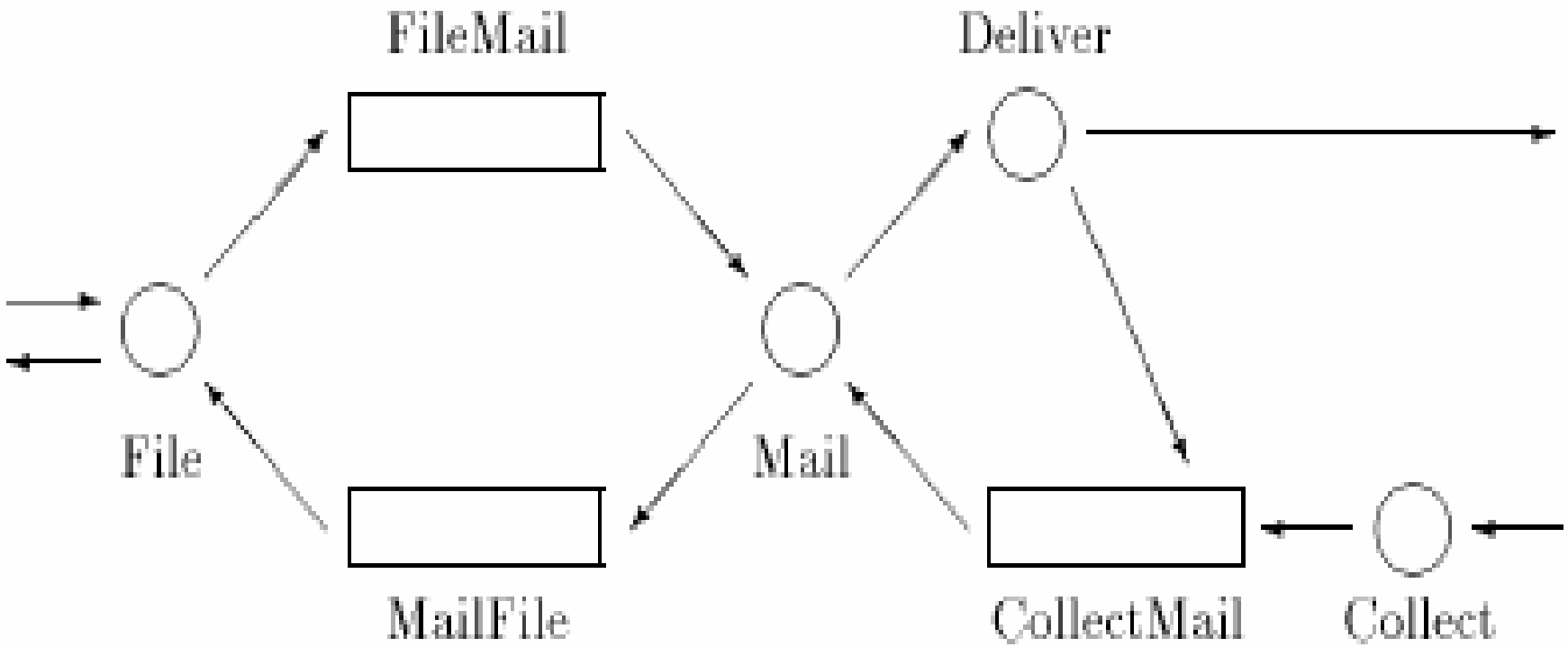
$$
\begin{aligned}
\mathrm{Mail} \quad &\stackrel{\mathrm{def}}{=} \quad \mathrm{fm\_remove} \,.\, (\tau \,.\, \overline{\mathrm{md\_start}} \,.\, \mathrm{md\_stop} \,.\, \mathrm{Mail1} \\
&\qquad\qquad\qquad\qquad + \; \tau \,.\, \overline{\mathrm{mf\_insert}} \,.\, \mathrm{Mail1}) \\
&\quad + \mathrm{cm\_remove} \,.\, \overline{\mathrm{mf\_insert}} \,.\, \mathrm{Mail1} \\
\mathrm{Mail1} \quad &\stackrel{\mathrm{def}}{=} \quad \mathrm{cm\_empty} \,.\, \mathrm{Mail} \; + \; \mathrm{cm\_remove} \,.\, \overline{\mathrm{mf\_insert}} \,.\, \mathrm{Mail1}
\end{aligned}
$$

Model CCS – proces File

- Wykonuje w pętli jedną z dwóch akcji:
  - Odbiera plik z sieci i umieszcza w buforze FileMail
  - Wyjmuje przesyłkę z bufora MailFile i wysyła ją

$$\mathrm{File} \stackrel{\mathrm{def}}{=} \mathrm{file\_received} \,.\, \overline{\mathrm{fm\_insert}} \,.\, \mathrm{File} + \mathrm{mf\_remove} \,.\, \overline{\mathrm{file\_send}} \,.\, \mathrm{File}$$

Model CCS



Dwa rodzaje zakleszczenia

$$
\begin{aligned}
\mathrm{Dead1} \stackrel{\mathrm{def}}{=} {} & (\overline{\mathrm{cm\_insert}}\,.\,\mathrm{Collect} \\
& \mid \mathrm{Deliver} \\
& \mid \overline{\mathrm{cm\_remove}}\,.\,\mathrm{CollectMail} \\
& \mid \overline{\mathrm{mf\_insert}}\,.\,\mathrm{Mail1} \\
& \mid \overline{\mathrm{mf\_remove}}\,.\,\mathrm{MailFile} \\
& \mid \overline{\mathrm{fm\_remove}}\,.\,\mathrm{FileMail} \\
& \mid \overline{\mathrm{fm\_insert}}\,.\,\mathrm{File}) \\
& \setminus \{\ldots\}
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{Dead2} \stackrel{\mathrm{def}}{=} {} & (\overline{\mathrm{cm\_insert}}\,.\,\mathrm{Collect} \\
& \mid \overline{\mathrm{cm\_insert}}\,.\,\overline{\mathrm{md\_stop}}\,.\,\mathrm{Deliver} \\
& \mid \overline{\mathrm{cm\_remove}}\,.\,\mathrm{CollectMail} \\
& \mid \mathrm{md\_stop}\,.\,\mathrm{Mail1} \\
& \mid \mathrm{MailFile} \\
& \mid \overline{\mathrm{fm\_remove}}\,.\,\mathrm{FileMail} \\
& \mid \overline{\mathrm{fm\_insert}}\,.\,\mathrm{File}) \\
& \setminus \{\ldots\}
\end{aligned}
$$

Eksperymenty

- Wyeliminowanie możliwości odbierania przesyłek z sieci – system bez zakleszczenia
- Wyeliminowanie możliwości wysyłania listów przez użytkownika – oba rodzaje zakleszczenia występują
- Rezygnacja z generacji przez proces Mail listu z opisem błędu – oba rodzaje zakleszczenia występują

Eksperymenty

- Rezygnacja z automatycznego przekierowywania przesyłek – występuje tylko zakleszczenie typu Dead1

$$\mathrm{Deliver} \stackrel{\mathrm{def}}{=} \mathrm{md\_start} \,.\, \overline{\mathrm{letter\_deliver}} \,.\, \overline{\mathrm{md\_stop}} \,.\, \mathrm{Deliver}$$

Eksperymenty

- Zmiana zachowania procesu Mail. Teraz nie opróżnia buforu CollectMail zanim zacznie obsługiwać bufor FileMail – te same możliwości zakleszczenia

$$
\begin{aligned}
\mathrm{Mail} &\stackrel{\mathrm{def}}{=} \mathrm{cm\_remove} \,.\, \mathrm{Mail1} + \mathrm{fm\_remove} \,.\, \mathrm{Mail2} \\
\mathrm{Mail1} &\stackrel{\mathrm{def}}{=} \overline{\mathrm{mf\_insert}} \,.\, (\mathrm{fm\_empty} \,.\, \mathrm{Mail} + \mathrm{fm\_remove} \,.\, \mathrm{Mail2}) \\
\mathrm{Mail2} &\stackrel{\mathrm{def}}{=} \tau \,.\, \overline{\mathrm{md\_start}} \,.\, \mathrm{md\_stop} \,.\, \mathrm{Mail3} + \tau \,.\, \overline{\mathrm{mf\_insert}} \,.\, \mathrm{Mail3} \\
\mathrm{Mail3} &\stackrel{\mathrm{def}}{=} \mathrm{cm\_empty} \,.\, \mathrm{Mail} + \mathrm{cm\_remove} \,.\, \mathrm{Mail1}
\end{aligned}
$$

Eksperymenty

- Modyfikacja procesu File. Teraz po otrzymaniu pliku z sieci, opróżnia bufor MailFile – występują te same możliwości zakleszczenia

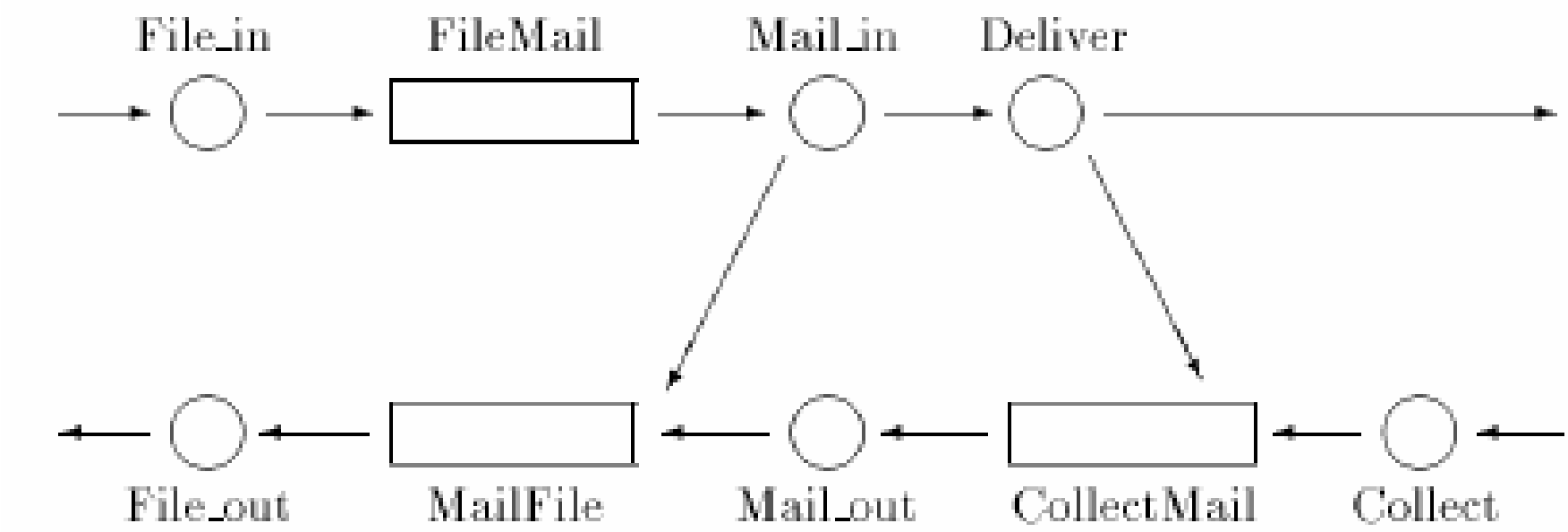
$$\mathrm{File} \stackrel{\mathrm{def}}{=} \mathrm{mf\_remove} \,.\, \overline{\mathrm{file\_send}} \,.\, \mathrm{File} + \mathrm{file\_received} \,.\, \overline{\mathrm{fm\_insert}} \,.\, \mathrm{File1}$$

$$\mathrm{File1} \stackrel{\mathrm{def}}{=} \mathrm{mf\_empty} \,.\, \mathrm{File} + \mathrm{mf\_remove} \,.\, \overline{\mathrm{file\_send}} \,.\, \mathrm{File1}$$

Wniosek

- Nie da się naprawić tego systemu poprzez samą zmianę kolejności wykonywania akcji
- Trzeba zmienić strukturę całego systemu:
  - Rozbicie procesu Mail na dwa – jeden opróżnia FileMail, a drugi CollectMail
  - Rozbicie procesu File na dwa – jeden odbiera pliki z sieci, a drugi opróżnia bufor MailFile

Nowy system


File_in
FileMail
Mail_in
Deliver
File_out
MailFile
Mail_out
CollectMail
Collect

Nowy system – model CCS

$$
\begin{aligned}
&(\mathrm{Collect} \\
&\quad \mid \mathrm{CollectMail} \\
&\quad \mid \mathrm{Deliver} \\
&\quad \mid \mathrm{Mail\_in} \mid \mathrm{Mail\_out} \\
&\quad \mid \mathrm{MailFile} \\
&\quad \mid \mathrm{FileMail} \\
&\quad \mid \mathrm{File\_in} \mid \mathrm{File\_out}) \\
&\setminus \{\mathrm{cm\_insert}, \mathrm{cm\_remove}, \mathrm{cm\_empty}, \\
&\quad \mathrm{md\_start}, \mathrm{md\_stop}, \\
&\quad \mathrm{mf\_insert}, \mathrm{mf\_remove}, \mathrm{mf\_empty}, \\
&\quad \mathrm{fm\_insert}, \mathrm{fm\_remove}, \mathrm{fm\_empty}\}
\end{aligned}
$$

Nowy system – model CCS

$$
\begin{aligned}
\mathrm{Mail\_in} &\stackrel{\mathrm{def}}{=} \mathrm{fm\_remove} \,.\, (\tau \,.\, \overline{\mathrm{md\_start}} \,.\, \mathrm{md\_stop} \,.\, \mathrm{Mail\_in} \\
&\qquad + \; \tau \,.\, \overline{\mathrm{mf\_insert}} \,.\, \mathrm{Mail\_in}) \\
\mathrm{Mail\_out} &\stackrel{\mathrm{def}}{=} \mathrm{cm\_remove} \,.\, \overline{\mathrm{mf\_insert}} \,.\, \mathrm{Mail\_out} \\
\mathrm{File\_in} &\stackrel{\mathrm{def}}{=} \mathrm{file\_received} \,.\, \overline{\mathrm{fm\_insert}} \,.\, \mathrm{File\_in} \\
\mathrm{File\_out} &\stackrel{\mathrm{def}}{=} \mathrm{mf\_remove} \,.\, \overline{\mathrm{file\_send}} \,.\, \mathrm{File\_out}
\end{aligned}
$$

Koniec