

# Lecture 10: Message Passing 2

Adam Hawley

February 27, 2019

## Contents

<b>1</b>	<b>Pascal-FC</b>	<b>1</b>
<b>2</b>	<b>Ada</b>	<b>1</b>
2.1	Ada Communication Model . . . . .	2
2.2	Other Facilities . . . . .	2
2.3	<code>select</code> . . . . .	2
2.3.1	<code>selective_accept</code> . . . . .	2
<b>3</b>	<b>Synchronisation</b>	<b>3</b>

## 1 Pascal-FC

- Synchronous communication
- Unlimited message types
- Indirect naming via channels
- Guarded select statements
- Has an extended rendezvous mechanism

## 2 Ada

- **Remote invocation** communication model
  - Can be used to provide **synchronous communication**
- Unlimited message types

- **Direct symmetric** naming via task names, and an entry defined for that task
- Guarded select statements
- Has extended rendezvous

## 2.1 Ada Communication Model

Based on a **client/server** coordination model. A **service** is a **public entry** in the server's task specification. An **entry** declaration specifies the name, parameters and result types for the service.

## 2.2 Other Facilities

- 'count gives the number of tasks queued on an entry.
- Entry families declare groups of entries
- Nested accept statements allow multiple task coordination
- A task executing in an **accept** can also execute an **entry** call

## 2.3 select

The select statement comes in four forms:

```
select_statement ::= selective_accept
                  conditional_entry_call
                  timed_entry_call
                  asynchronous_select
```

### 2.3.1 selective\_accept

This allows the server to:

- wait (for more than one more rendezvous at a time)
- time-out (if no rendezvous occurs within a specified time)
- terminate (if no client can ever call an entry)

### 3 Synchronisation

- Both tasks must *agree* to communicate
- **Ready** task **waits** for the other task (blocked)
- When both tasks are ready, client's arguments are passed to the server.
- Server executed code inside the **accept** statement
- Results returned to client at completion of **accept**
- Tasks continue concurrently

I can't tell if there was actually more content or not...