Norges teknisk–naturvitenskapelige universitet
Institutt for datateknikk og informasjonsvitenskap

TDT4105 IT Grunnkurs
Høst 2014

Løsningsforslag — Øving 9

1 Teori

a) En protokoll er et regelverk som bestemmer hvordan kommunikasjon skal foregå, og hvilke funksjoner som kan brukes.

b) HTTP er en forkortelse for Hypertext Transfer Protocol. Det er en protokoll for overføring av tekst og bilder over Internet. Protokollen brukes i hovedsak av nettlesere og webservere til å utveksle nettsider.

c) Applikasjonslaget, transportlaget, nettverkslaget, linklaget og det fysiske laget.

2 Strukturer

a)

```
date = struct('day', 20, 'month', 5, 'year', 1990);
```

b)

```
function print_date( date )
  fprintf('%02i.%02i.%i', date.day, date.month, date.year);
end
```

c)

```
person = struct('name', 'Per', 'date_of_birth', struct('day', 20, '
                                        month', 5, 'year', 1990), '
                                        phone', 48151623);
```

d)

```
function print_person( person )
  fprintf('%s, ', person.name);
  print_date(person.date_of_birth);
  fprintf(', %i, ', person.phone);
end
```

e)

```
function person = prompt_person()
  name = input('Hva heter du? ', 's');
  day = input('Hvilken dato er du foedt paa? ');
  month = input('Hvilken maaned er du foedt i? ');
  year = input('Hvilket aar er du foedt i? ');
  phone = input('Hva er telefonnummeret ditt? ');

  date_of_birth = struct('day', day, 'month', month, 'year', year);
  person = struct('name', name, 'date_of_birth', date_of_birth, '
                                          phone', phone);
end
```

f)

```
function age = get_age( person )

    day = person.date_of_birth.day;
    month = person.date_of_birth.month;
    year = person.date_of_birth.year;

    [Y M D] = datevec(now);
    age = Y - year;

    if M < month || (M == month && D < day)
        age = age - 1;
    end
end
```

g)

```
function persons = batch_register_persons()
  i = 1;
  while true
    persons(i) = prompt_person();
    answer = input('Skal du registrere flere personer (ja/nei)? ', '
                                        s');

    if strcmp(answer, 'nei')
      break;
    end

    i = i + 1;
  end
end
```

h)

```
function list_persons( list_of_persons )

  for i = 1:length(list_of_persons)
    print_person(list_of_persons(i));
    fprintf('\n');
  end

end
```

3 Persondatabase

a)

```
function serialized = serialize_date( date )

  d = int2str(date.day);
  m = int2str(date.month);
  y = int2str(date.year);

  if date.day < 10
    d = strcat('0', d);
  end

  if date.month < 10
    m = strcat('0', m);
  end

  serialized = strcat(d, '.', m, '.', y);

end
```

```
function serialized = serialize_person( person )

  n = person.name;
  d = serialize_date(person.date_of_birth);
  p = int2str(person.phone);

  serialized = strcat(n, '#', d, '#', p);

end
```

b)

```
function deserialized = deserialize_date( date )

  [tok rest] = strtok(date, '.');
  d = str2num(tok);
  [tok rest] = strtok(rest, '.');
  m = str2num(tok);
  tok = strtok(rest, '.');
  y = str2num(tok);

  deserialized = struct('day', d, 'month', m, 'year', y);

end
```

```
function deserialized = deserialize_person( person )

  [tok rest] = strtok(person, '#');
  n = tok;
  [tok rest] = strtok(rest, '#');
  d = deserialize_date(tok);
  tok = strtok(rest, '#');
  p = str2num(tok);

  deserialized =struct('name', n, 'date_of_birth', d, 'phone', p);

end
```

c)

```
function store( filename, persons )

  fileID = fopen(filename, 'w');

  for p = persons
    fprintf(fileID, '%s\n', serialize_person(p));
  end

  fclose(fileID);

end
```

d)

```
function persons = loadfile( filename )

  persons = [];

  fileID = fopen(filename, 'r');

  line = fgets(fileID);
  while ischar(line)
    person = deserialize_person(line);
    persons = [persons person];
    line = fgets(fileID);
  end

  fclose(fileID);

end
```

4 Personprogram

a)

```
function person_program()

  choice = -1;
  P = [];

  print_header();
  while choice ~= 0
    print_menu();
    choice = input('Velg et tall: ');
    if choice == 0
      continue;
    end
    P = process(P, choice);
  end
  print_footer();
end
```

b)

```
function print_header()
  fprintf('Velkommen til persondatabasen\n');
end
```

c)

```
function print_menu()

  fprintf('%s\n', '1. Hent database fra fil');
  fprintf('%s\n', '2. Lagre database til fil');
  fprintf('%s\n', '3. List alle personer');
  fprintf('%s\n', '4. Legg inn ny person');
  fprintf('%s\n', '5. Endre person');
  fprintf('%s\n', '0. Avslutt programmet');

end
```

d)

```
function print_footer()
  fprintf('Programmet avsluttes.\n');
end
```

e)

```
function P = process_load()
  fprintf('Skal laste en fil\n');
end

function process_store(P)
  fprintf('Skal lagre en fil\n');
end

function process_list_persons(P)
  fprintf('Skal skrive ut personene\n');
end

function P = process_new_person(P)
  fprintf('Skal legge til en ny person\n');
end

function P = process_change_person(P)
  fprintf('Skal endre en person\n');
end
```

f)

```
function P = process(P, choice )

  fprintf('\n');
  switch choice
    case 1
      P = process_load();
    case 2
      process_store(P);
    case 3
      process_list_persons(P);
    case 4
      P = process_new_person(P);
    case 5
      P = process_change_person(P);
    otherwise
      fprintf('%s\n', 'Ugyldig valg');
  end
  fprintf('\n');
end
```

g)

```
function P = process_load()

  filename = input('Hvilken fil vil du laste? ', 's');
  P = loadfile(filename);

end
```

h)

```
function process_store(P)

  filename = input('Hvilken fil vil du lagre til? ', 's');
  store(filename, P);

end
```

i)

```
function process_list_persons(P)

  list_persons(P);

end
```

j)

```
function P = process_new_person( P )

  P = [P prompt_person()];

end
```

k)

```
function P = process_change_person(P)

  index = input('Hvilken person vil du endre? ');
  P(index) = prompt_person();

end
```