CSCI 3753: Operating Systems Fall 2024

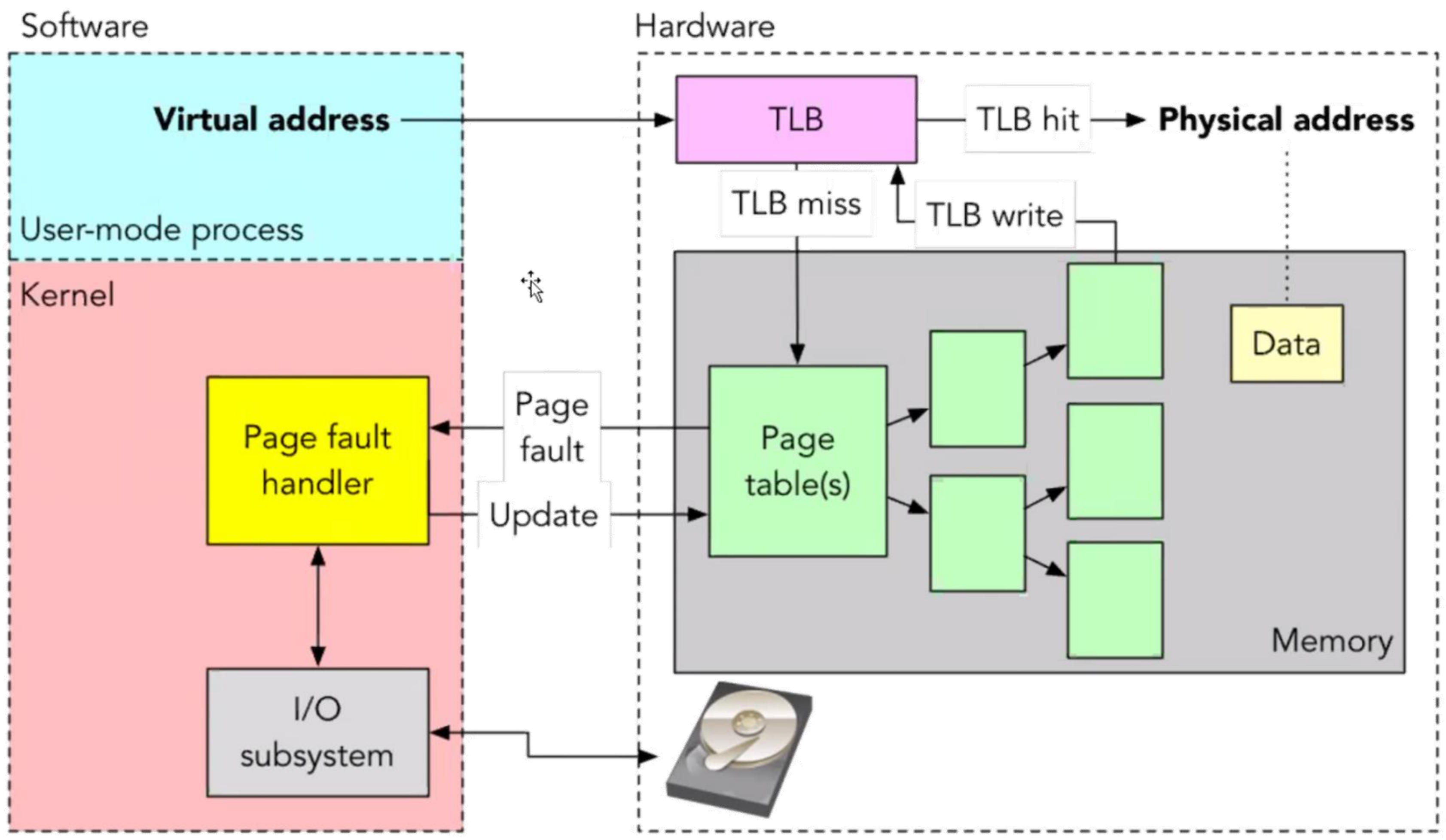
Dylan Sain

Department of Computer Science

University of Colorado Boulder

Week 13: Program Assignment 8 Prediction

Paging Simulator



Paging Simulator

- Goal:

Implement a paging strategy that a paging simulator can use to maximize the performance of the memory access in a set of pre-defined programs

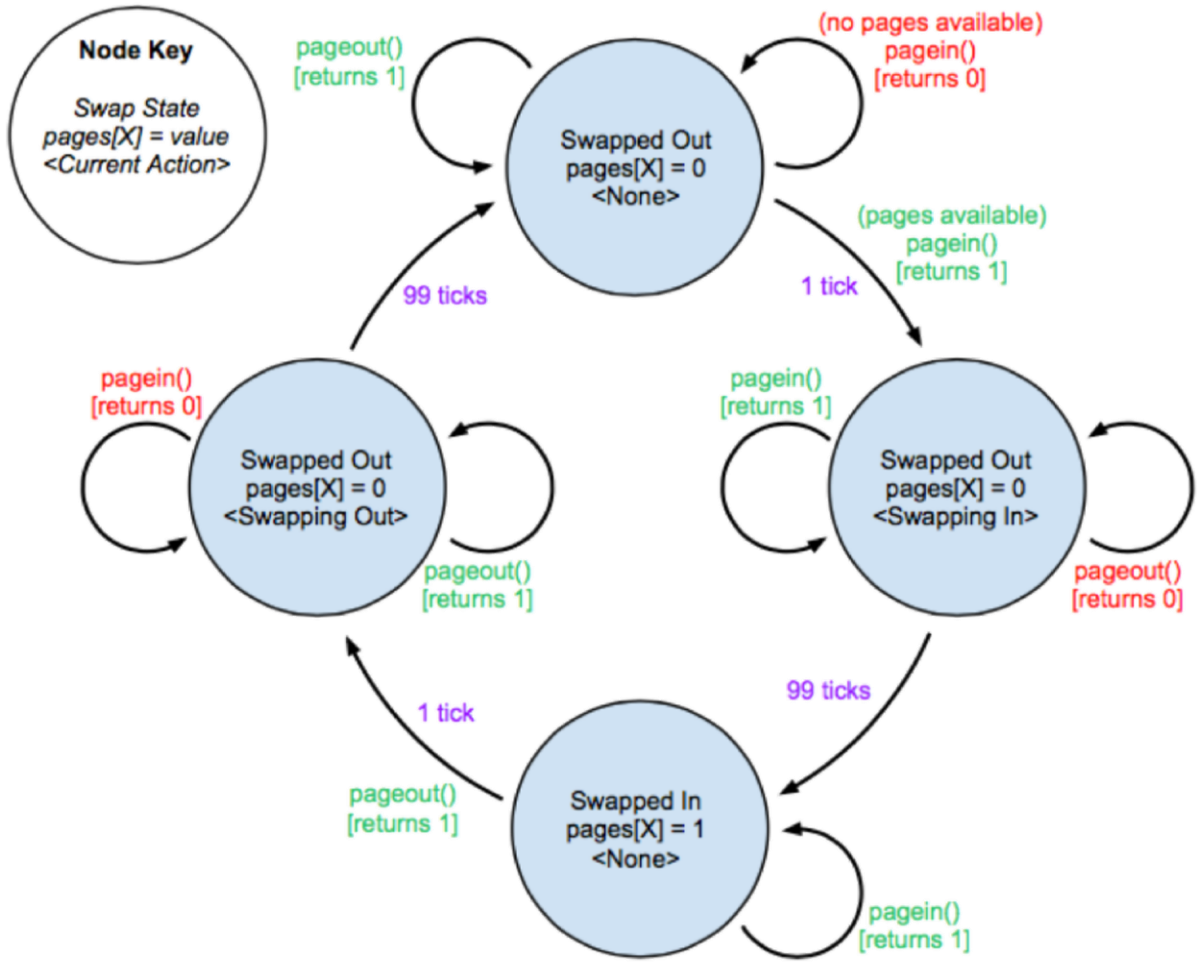
- Default values:
  - 10 virtual pages per process (MAXPROCPAGES)
  - 20 simultaneous processes competing for pages (MAXPROCESSES)
  - 50 physical pages (frames) in total (PHYSICALPAGES)
  - 100 tick delay to swap a page in or out (PAGEWAIT)
  - 256 memory unit page size (PAGESIZE)
  - 40 processes run in total (QUEUESIZE)

Paging Simulator

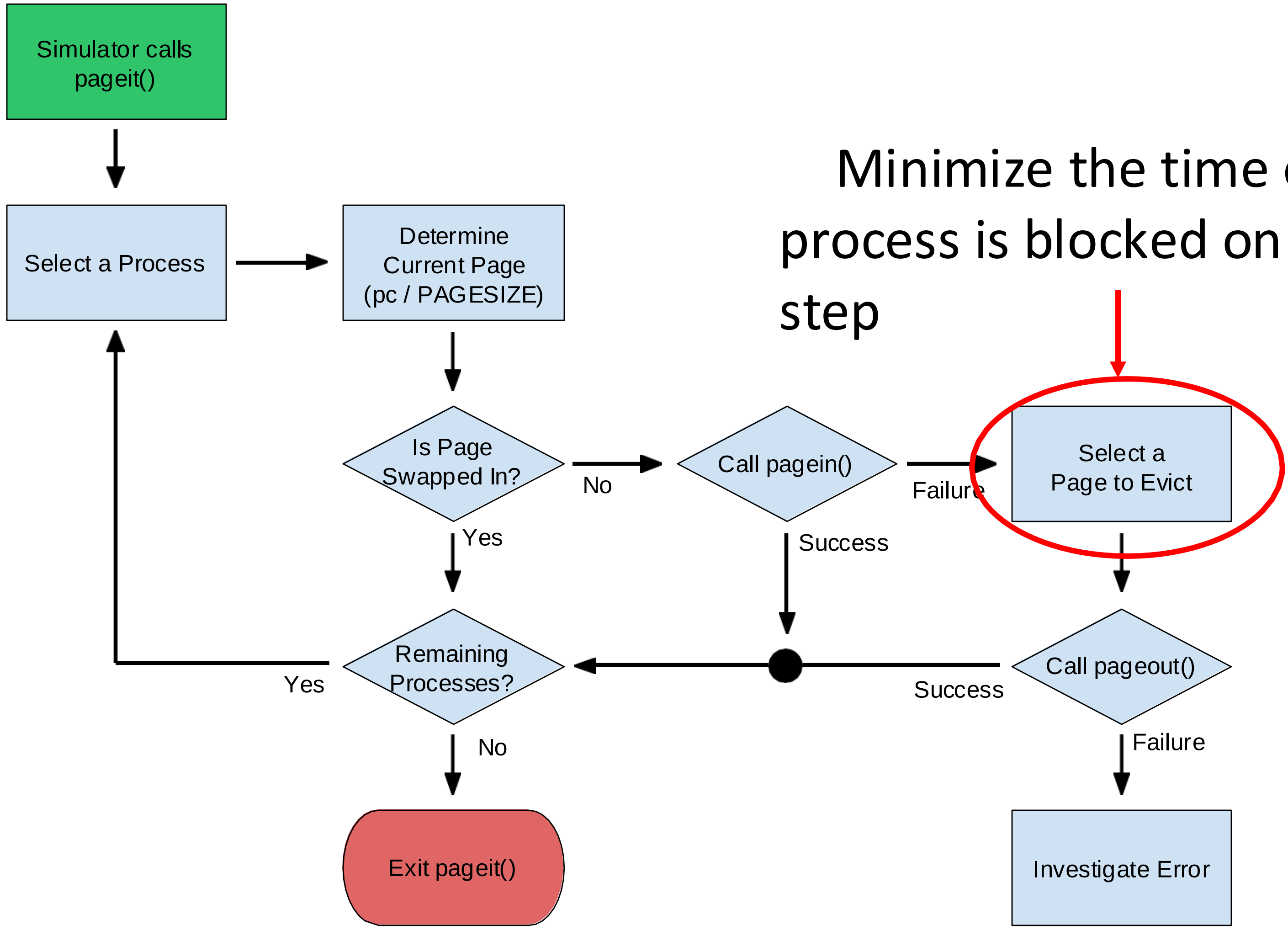
- Key functions for interaction
  - To control the allocation of virtual and physical pages
    - pagein()
    - pageout()
  - To handle the page fault
    - pageit() ☒ core paging function that needs implementation

- Action items
  - Implement any form of predictive paging algorithm (PA8): pager-predict.c

Possible Page States and Transitions

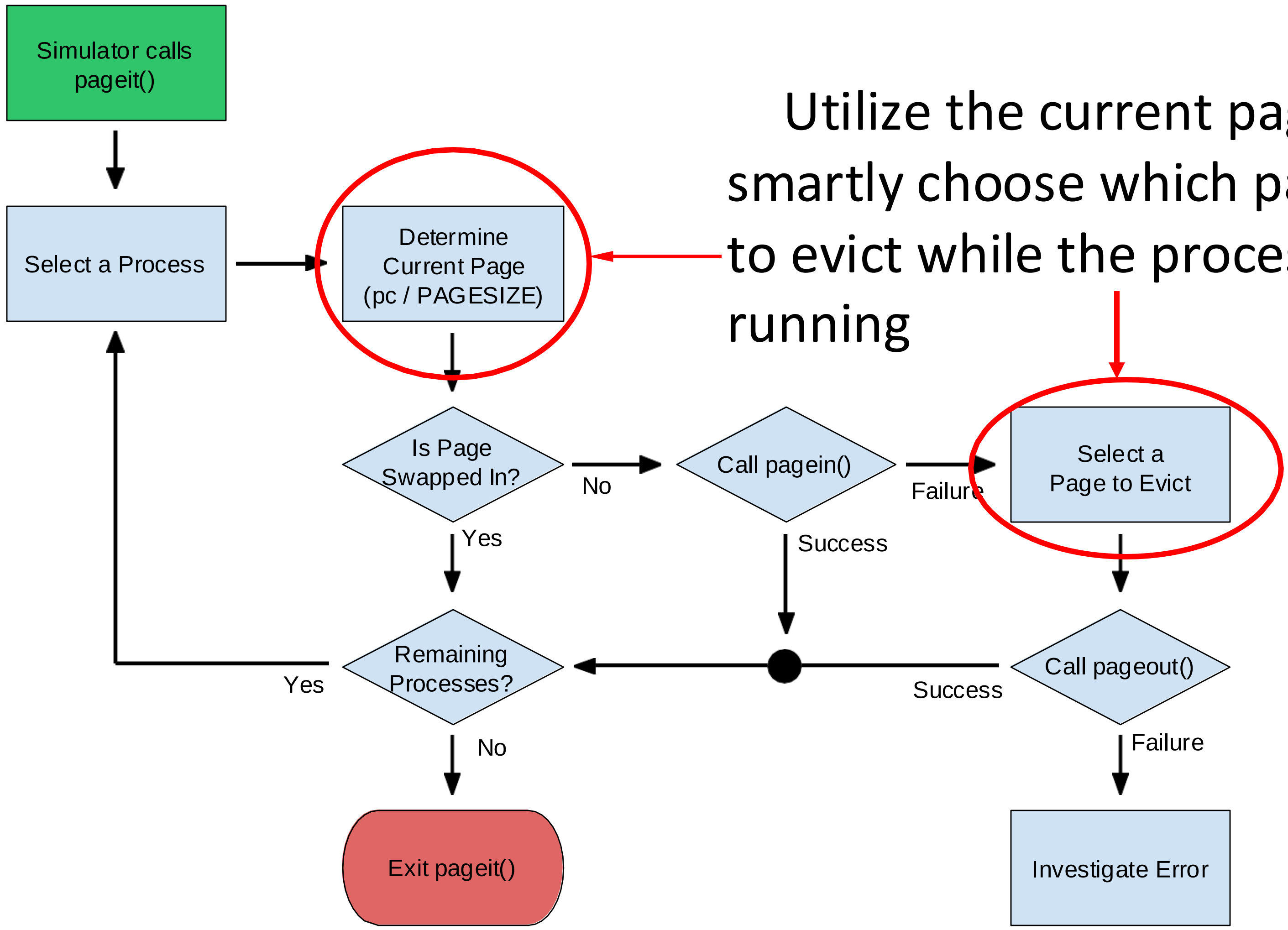


Why Predictive Paging



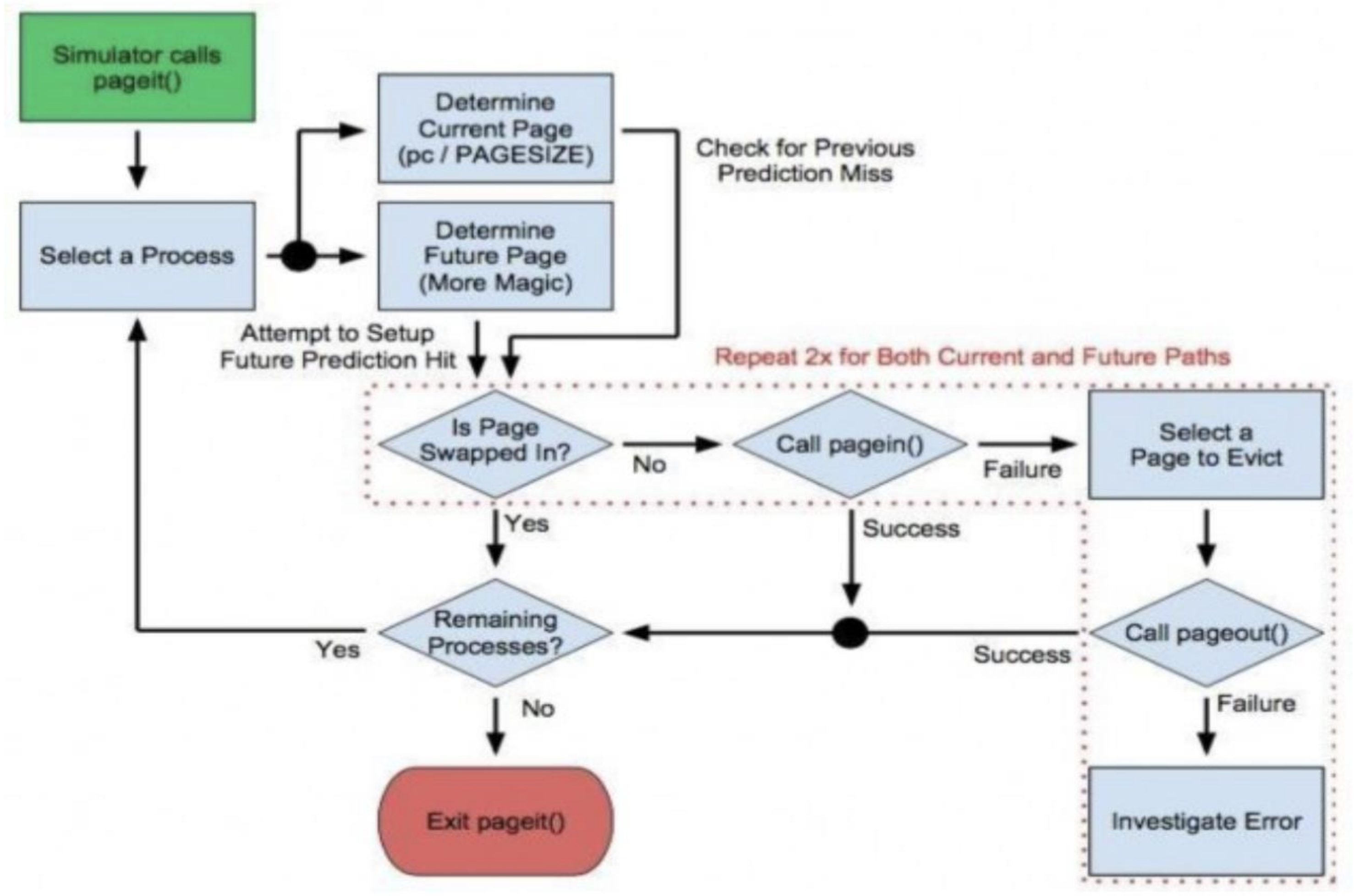
Minimize the time each process is blocked on this step

How Predictive Paging



Utilize the current page to smartly choose which pages to evict while the process is running

Updated diagram



Implementation Ideas

- Use the program counter (PC) to identify loops and patterns
    - Once in a pattern assume pattern will repeat and page in the respective pages
    - Ex: 1 -> 4 -> 2 -> 6 -> 5 -> 1 -> ….
        - Assume page size of 3
        - If on execution 4 preload 2 and 6 sense we know the pattern will continue and load these next.
- With the PC and prior knowledge of the programs to identify a program and preload for that program
    - Must understand the types of programs to accurately predict next steps

General Strategies

- Keep track of the PC at all times!
  - Very important to identify which program or find patterns and loops
- Keep track of which pages are being paged in or out
  - Useful for figuring out which pages to page in next
- Identify all page faults and predictive misses!
  - Good to figure out how to optimize your program.

Types of Programs

Program 1 - A loop with an inner branch

```
# loop with inner branch
for 10 30
  run 500
  if .4
    run 900
  else
    run 131
  endif
end
exit
```

Program 5 - Probabilistic backward branch

```
# probabilistic backward branch
for 10 20
  label :
    run 500
    if .5
      goto label
    endif
end
exit
```

Program 2 - Single loop

```
# one loop
for 20 50
 run 1129
end
exit
```

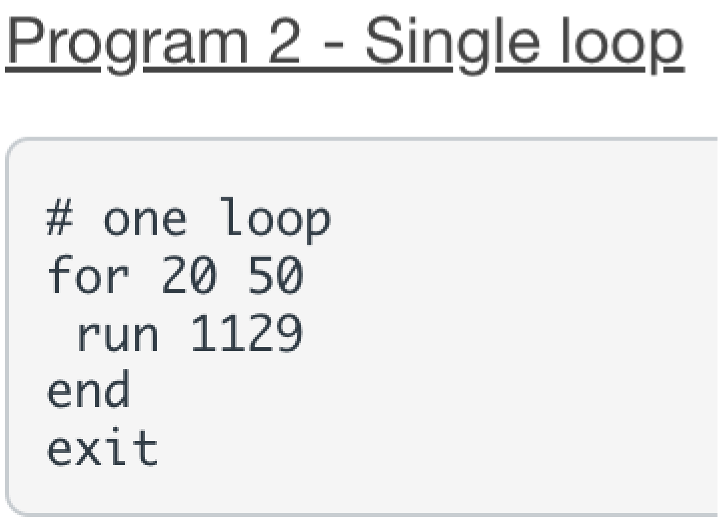
Program 3 - Double nested loop

```
#doubly-nested loop
for 10 20
  run 1166
  for 10 20
    run 516
  end
end
exit
```

Program 4 - Linear

```
#entirely linear
run 1911
exit
```

Program 2 Ideas



Program 2 - Single loop

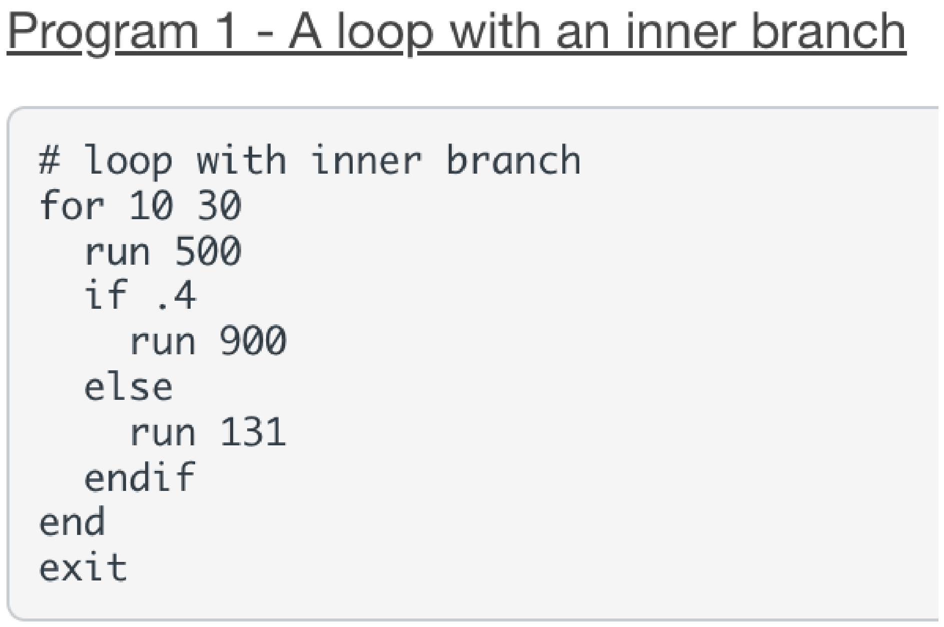
```
# one loop
for 20 50
 run 1129
end
exit
```

- Identify the loop by watching the PC go back to the start and request the same pages
- Identify the pages being used in this loop
- Start counting the number of times the loop has been run
- Start pre-paging during execution for the next pages
- After 20 iterations use less pre-paging during the end of the loop to prepare for end of loop cycle (50 = done)

Activity: Identify strategies and potential problems

As a group let's break down the various programs and identify ways to use predictive paging

Program 1 Ideas: Group ideas

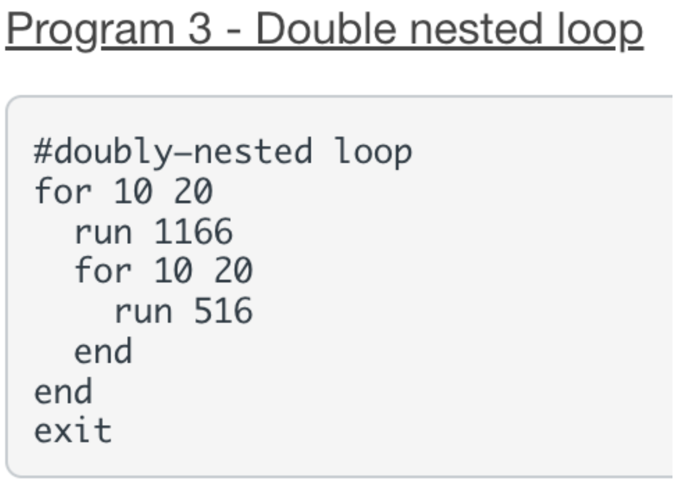


Program 1 - A loop with an inner branch

```
# loop with inner branch
for 10 30
  run 500
  if .4
    run 900
  else
    run 131
  endif
end
exit
```

- It can run either 900 or 131 (131 more likely load first)
- Load 2 pages for 131 and only 1 for 900
- When you loop always return to same spot but two different program counter to go through
- Same random number loop

Program 3 Ideas

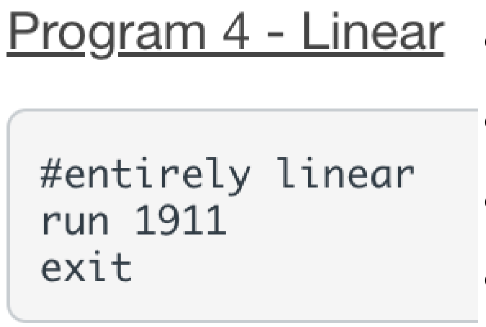


Program 3 - Double nested loop

```
#doubly-nested loop
for 10 20
  run 1166
  for 10 20
    run 516
  end
end
exit
```

- Outer and inner loop both run random number of times
- Pre-paging for both 1166 and 516 after 10 inner loop iterations
- Doesn't loop to the start IMPORTANT

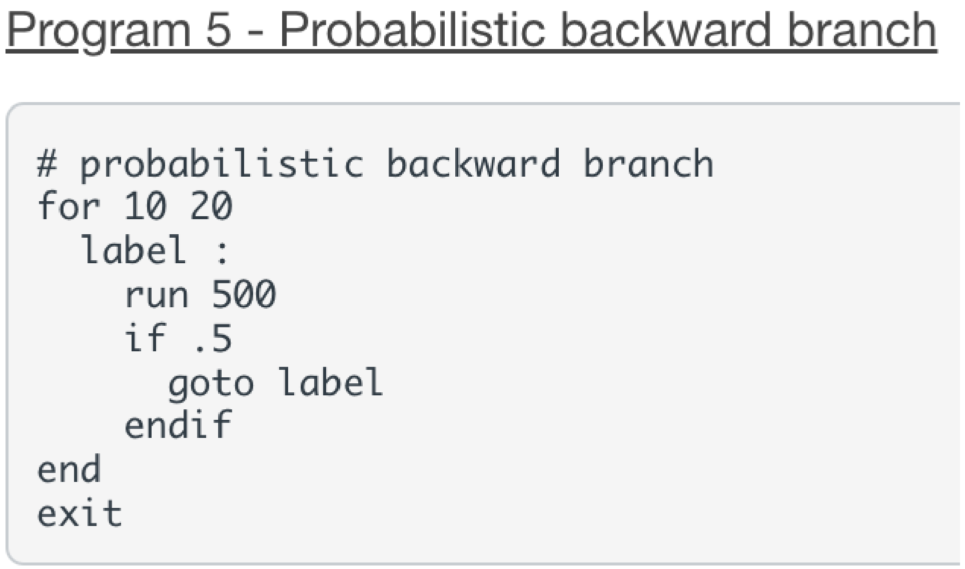
Program 4 Ideas


Program 4 - Linear

```
#entirely linear
run 1911
exit
```

- No looping structure to identify the program
- Hard to identify
- Keep track of pages that are loaded in
- Good program to take a look at the markov chains and transition matrix
- Simplest program but hardest to predict

Program 5 Ideas



Program 5 - Probabilistic backward branch

```
# probabilistic backward branch
for 10 20
  label :
    run 500
    if .5
      goto label
    endif
end
exit
```

- 50 50 chance to go back to beginning of loop without incrementing the counter
- Always pre-page run 500
- The PC might be able to tell you if you jumped or if you looped
- Don’t known when the loop has finished